AD-A039 761 KANSAS UNIV/CENTER FOR RESEARCH INC LAWRENCE F/G 17/2
VIDEO BANDWIDTH COMPRESSION.(U)
FEB 77 N C GRISWOLD, R M HARALICK F33615-74-C-1093
UNCLASSIFIED 257-4 AFAL-TR-76-102 NL

1 OF 3
AD
A039761

AFAL-TR-76-102

VIDEO BANDWIDTH COMPRESSION

THE UNIVERSITY OF KANSAS
CENTER FOR RESEARCH, INC.
LAWRENCE, KANSAS 66044

FEBRUARY 1977

FINAL REPORT DECEMBER 1973 - NOVEMBER 1976

Approved for public release; distribution unlimited

AIR FORCE AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

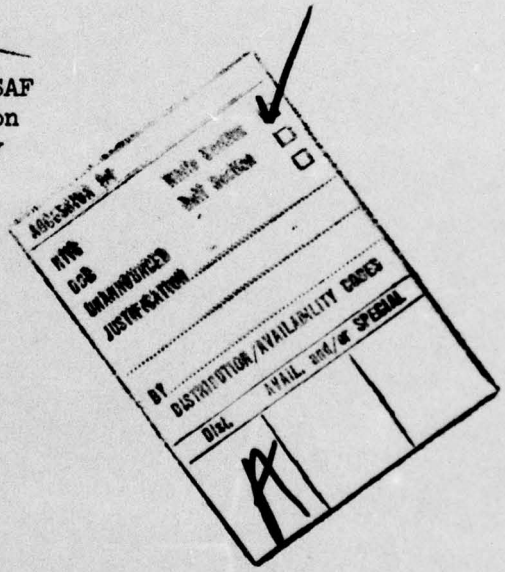
Louis A. Tamburino
LOUIS A. TAMBURINO
Project Engineer

John O. Mysing
JOHN O. MYSING, Tech. Mgr.
Information Presentation &
Control Group
System Technology Branch

FOR THE COMMANDER

James D. Everett
JAMES D. EVERETT, Colonel, USAF
Chief System Avionics Division
Air Force Avionics Laboratory

AIR FORCE - 30 MARCH 77 - 50



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFAL-TR-76-102	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Video Bandwidth Compression	5. TYPE OF REPORT & PERIOD COVERED: Dec. 1973 to Final report Dec. 1975
6. PERFORMING ORG. REPORT NUMBER: 257-4	
7. AUTHOR(s): Norman C. Griswold, Robert M. Haralick	8. CONTRACT OR GRANT NUMBER(s): F33615-74-C-1093
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Kansas/Center for Research, Inc., Irving Hill Road, Campus West, Lawrence, KS 66044	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62204F, 2003-02-15
11. CONTROLLING OFFICE NAME AND ADDRESS: Systems Tech Branch, Air Force Avionics Laboratory, Wright Patterson Air Force Base, Ohio 45433	12. REPORT DATE: Feb[redacted]1977
	13. NUMBER OF PAGES: 232
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Final rept. Dec 73 - Dec 75.	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release;
Distribution unlimited

245 p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

1. Reconnaisance
2. Data Link
3. Video
4. Bandwidth Compression
5. Remotely Piloted Vehicles
6. Digital Image Processing
7. Fast Transforms

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

400 874

20. ABSTRACT

This project report summarizes the bandwidth compression research activities performed by the University of Kansas under contract number F33-615-74-R-1093 with the Air Force Avionics Laboratory at Wright Patterson Air Force Base, Ohio.

The primary purpose of this study is to investigate the feasibility of video bandwidth compression in the order of 50:1. This compression was simulated using imagery digitized to 64 grey levels (6 bits). This task was accomplished by three steps. These steps are: transform compression, frame rate reduction, and Differential Pulse Code Modulation (DPCM). Since the transform compression scheme is vital to the success of this project, major emphasis was placed on this area. Results indicate that a 50:1 compression is feasible and that the best transforms to be utilized in a hybrid manner with DPCM is the Discrete Cosine. Additional research developments led to a fast implementation of the Karhunen Loeve transform which is optimum under root mean square error criteria. Comparisons of other fast transform performance is made as well as an optimum bit coding scheme.

PREFACE

In the selection of the subject of this analysis, Bandwidth Compression of Video Signals from Remotely-Piloted Vehicles (RPV Systems), consideration was given to the importance of the topic itself to the Air Force and Avionics Community. Many Air Force vehicles and satellites act as remote sensing platforms for collecting weather data, reconnaissance, earth resources studies and television coverage of the moon's surface. The Armed Services, and especially the Air Force, are tending toward an all-digital aircraft. These future plans rely upon the technological advancements and research time required to solve many problems associated with digital avionics. All of these applications may be analyzed by the use of spectral and/or spatial data of remotely sensed scenes. It is of prime importance to consider not only what data can be collected, but what data must be transmitted, at what rate the transmission must take place, and the resolution requirements of the reproduced data.

These several applications have many common problems. The vast amounts of data require large bandwidth systems or long transmission times at much narrower bandwidths, excessive memory constraints and excessive analysis time. Standard commercial TV frame rates of 30 frames/sec. result in a nominal 4.5 MHz analog bandwidth with a transmission rate of about 20 Megabits/sec. With the cost-driven system constraints, it is imperative that only minimum information be transmitted. The redundancy within the sensed environment must be capitalized upon and the size and weight constraints of computerized systems must be held to an absolute minimum. Applications of appropriate bandwidth compression techniques can have a major impact in cost and efficiency of transmission systems for the Air Force. As applied to RPV systems, two major considerations must be investigated: First, the reconnaissance role in which high resolution data must be preserved, and second, the near real

time transmission problem under flight. In the former, it is imperative that the information content and resolution requirements be maintained but may not require rapid analysis, while memory storage requirements are the forcing function. In the latter, near real time suggests a relaxation of resolution constraints but transmission speed and simplicity are of prime concern. The need to define these transform parameters and their limitations is emphasized by the Control Data Retrieval Systems Working Group at Wright-Patterson AFB.

ACKNOWLEDGMENTS

The author would like to acknowledge those who have encouraged, worked with, and spent many sleepless nights to make this program a success. I would first acknowledge my wife and family whose support through this program made the most difficult times bearable. A wife who believes in education, total commitment, and never complained about the many hours spent alone while being both mother and father to our children shares in the accomplishment of this task.

To the team which made the management task simple and shared the many nights of long data processing I owe the technical success of this program. Gary Minden, Craig Paul, and Nimit Kattiyakulwanich have each given all their loyalty, and technical ability to the performance of each task assigned to them. Their accomplishments in applying the theory to operating software has been enormous. Each deserves special recognition for all the software packages but in particular for their major contributions listed below.

Gary Minden for the accomplishment of implementing the bit allocation programs and the Kandidats transform package.

Craig Paul for the standard Karhunen Loeve and Fast K-L transform package.

Nimit for the implementation of the transforms, probability and correlation packages as well as the initial Fast K-L programs.

I especially thank all my professors who instructed me during my residence at the University of Kansas. To Dr. Haralick, my advisor, I owe so much. He has taught me the meaning of a significant accomplishment and guided our technical accomplishments with patience. He has renewed with me the need to know and to accomplish.

Special acknowledgment is also given to the Air Force Avionics Lab for support of this project and to the CRINC staff who have supported the many reports, drafting and photography. Special thanks is given to Mrs. Ruth Hatfield for typing this manuscript.

I further wish to acknowledge Dr. K. Sam Shanmugam, of Wichita State who supplied the necessary notes and changes for the optimum bit allocation derivation and programming.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

SECTION I
TRANSFORM IMAGE DATA COMPRESSION

1. INTRODUCTION

In the design of image coding systems for digital transmission of images one is faced with providing a method which can be reasonably implemented to minimize the number of digital bits used for transmission and keep distortions of the original scene within a pre-determined fidelity criterion. With the power of digital computers and the introduction of fast transforms, the transform of the image is encoded rather than the image itself. This form of data requires long transmission times at a fixed bit rate determined by the design. It is therefore advantageous to reduce or compress the data while preserving the information content. In remote sensing applications from satellites or remotely piloted vehicles, the data contains a large amount of redundant information due to high correlation of graytones of spatially adjacent resolution cells. It is this property which permits bandwidth compression of digital images. It is therefore desirable in terms of transmission time, required storage, and as a preventative to signal jamming when used with spread spectrum techniques to achieve bandwidth compression.

Briefly, bandwidth compression schemes fall into three categories[1]:

1. Statistical models
2. Psychovisual models
3. Transform techniques

The statistical models appear at first glance to incorporate the natural randomness of data. However, actual probabilities of data points within some general set are not often known accurately. Mean and variance are gross measures and do not lend themselves easily to picture coding in terms of reconnaissance value. Schrieber has investigated third order

statistics but sufficient data have not yet been obtained [1]. The psychovisual approach accounts for the inability of the receiver, the human eye, to detect signals beyond certain gray limits while being slightly affected by high frequency phenomenon even though the magnitude of such disturbance may be low. Again, however, modeling the eye or the human responses is not complete and is plagued with non-linearities still to be accounted for. The transform technique appears to offer an approach which can include both effects of the above and lend itself to digital modeling [1].

This research applies the transform technique to bandwidth compression of remotely piloted vehicles. In this report, the concepts of bandwidth compression are developed. The performance criteria is presented with the procedure for the Karhunen-Loeve transform which is optimum under the mean square error criteria. As part of the conducted research, a fast Karhunen-Loeve procedure is developed. The assumptions for the separability of this transform are presented with data demonstrating the optimality when compared to other fast transforms. With regard to the coding for compression, three procedures are used: the notch filter, energy compression, and a variable word length code. The difference between the fast transforms which are feasible for interframe processing are discussed with the conclusion that with this variable word length code the Discrete Cosine is the best performer. How this selection was made is reviewed in light of the established criteria and sequency properties of the fast transforms.

The feasibility of a system compression of 50:1 is then investigated. This is done by the application of the "best" transform with Differential Pulse Code Modulation (DPCM) in the frequency domain between frames. An additional analysis of the differences which contribute new information for updating has been completed.

This research therefore offers evidence of the feasibility of system compression of 50:1 and the errors incurred in this process as well as complete computer documentation of programs directly generated for the compression scheme.

2. Review of the Literature

The principles of redundancy reduction, or data compression, have surprisingly, been known for some time. Its application, in image data compression has, over the past several years, sparked intense investigation. As early as 1958, Good [2] described a technique of eliminating redundancy in the matrix transform by matrix factorization. Fourier Analysis and other orthogonal transforms were well defined but decomposition was not to have its impact until 1965. Cooley and Tukey [3] reported a computationally efficient means to calculate the complex Fourier coefficients. With the power of digital computing the door was now open to image transmission and the investigation into comparative approaches. By March 1967, The Proceedings of the IEEE presented a collection of articles under a "Special Issue on Redundancy Reduction".

Andrews and Pratt [4] used an approach which treated the Fourier Transform of complete pictures. Anderson and Huang [5] followed with an adaptive version. In this approach the image was divided into a number of subimages and the coefficients selected for transmission was based on a linear relationship to the standard deviation. This became known as the Piecewise Fourier method and when used with adaptive thresholding reportedly performed well in the presence of noise. The discrete Fourier transform for an image is given by Andrews [6] and other texts as:

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp\left\{\frac{-2\pi i}{N}(xu+yv)\right\}$$

The inverse Fourier for reconstruction is:

$$f(x,y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \exp \left\{ \frac{2\pi i}{N} (ux+vy) \right\}$$

Many researchers have transmitted the complete Fourier transform of the picture. This particular process begins by the partitioning of an N x N picture into $(N/n)^2$ (n x n) sub-pictures where $n \ll N$. Having sampled the image in terms of brightness the discrete Fourier transform of each sub-picture is expanded into its frequency coefficients and an adaptive threshold procedure is applied.

Huang [5] defines a quantity L which is linearly proportional to the integer part of σ the second moment of the sub-picture defined as:

$$\sigma = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} (\beta_{m,n} - \alpha)^2 \quad 1/2$$

α is the ensemble average which is used as an estimate of the true mean and given by:

$$\alpha = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \beta_{m,n}$$

$\beta(m,n)$ is brightness values of the two-dimensional sub-picture. L coefficients are transmitted with this method. To express the Fourier transform as a matrix let $W = \exp(\frac{-2\pi i}{N})$ then $[F] = [A]\,[f]\,[A]$. Since the Fourier is a symmetrical and separable matrix:

$$[A] = \frac{1}{\sqrt{N}} \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ . \\ . \\ . \\ N-1 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 0 & 1 & 2 & . & . & . & . & N-1 \end{array} \\ \begin{bmatrix} W^0 & W^0 & W^0 & . & . & . & . & W^0 \\ W^0 & W^1 & W^2 & W^3 & . & . & . & W^{N-1} \\ W^0 & W^2 & & & & & & \\ . & W^3 & & & & & & \\ . & . & & & & & & \\ . & . & & & & & & \\ W^0 & W^{N-1} & . & . & . & . & . & W^{(N-1)} \end{bmatrix} \\ u \longrightarrow \end{array} \quad x \downarrow$$

$$W^{ux} = W^{ux \bmod N}$$

The Hadamard Approach:

The Hadamard transform is based on the Walsh functions. Walsh functions can be generated by first writing the binary representation of a word, converting to a gray code, and then multiplying the Rademacher functions represented by a 1 and leaving out those represented by a zero [7]. Walsh function properties are defined by Lackey [7]. The matrix equation is a good way to represent the operation with Walsh functions. If [X] is a column vector of our sampled brightness values, [W] a matrix of Walsh functions, then the Walsh coefficients are given by $[A] = [W] \cdot [X]$. Since the Walsh matrix can be classified as separable and symmetric, the Walsh matrix is its own inverse with a constant multiplier N. i.e.,

$$[X] = \frac{1}{N} \cdot [W] \cdot [A]$$

$$[W] \cdot [W] \cdot \frac{1}{N} = [I]$$

The Hadamard matrix has this same property [8]:

$$[H]\ [H]^t = N[I]$$

or for symmetric case $[H]\ [H] = N[I]$.

The lowest ordered Hadamard matrix defined by Pratt & Andrews [6] is $[H] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

Constructions for nearly all values of order N exist up to $2<N<200$. If $N = 2^n$ n = integer, H is a matrix of order N then $G = \begin{bmatrix} H & H \\ H & -H \end{bmatrix}$ is a Hadamard matrix of order 2N [6]. The frequency representation of the transform is given by its sequency coined by Harmuth. Sequency designates the number of sign changes as in a Rademacher function. The two-dimensional matrix representation of the transform pair is given by:

$$[F(u,v) = [H(u,v)]\ [f(x,y)]\ [H(u,v)]$$

$$[f(x,y)] = \frac{1}{N^2}\ [H(u,v)]\ [F(u,v)]\ [H(u,v)]$$

And the series for two dimensions is:

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)(-1)^{p(x,y,u,v)}$$

$$p(x,y,u,v) = \sum_{j=0}^{n-1} (u_i x_i + v_i y_i)$$

u_i, v_i, x_i, y_i are binary terms for u, v, x, y.

The transform possesses the same properties of the Fourier in terms of resistance or isolation to channel noise, the conservation of energy holds (Parseval's relation):

$$\text{i.e., } \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |f(x,y)|^2 = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |F(u,v)|^2$$

and has less computations. Since images generally do not follow a square wave pattern, more inherent mean square error is produced using this transform. A trade-off of allowable error and speed should always be considered. The Fourier operations can be reduced to $2N \log_2 N$ basic operations which is defined as a complex multiplication followed by a complex addition. The Hadamard on the other hand can be accomplished with $2N \log_2 N$ additions.

Discrete Linear Basis Technique:

The discrete linear basis technique is a new transform developed by R. M. Haralick at the Remote Sensing Laboratory, University of Kansas [9]. This technique makes use of a set of basis vectors independent of the data. Once the image has been divided into subimages the n^2 dimensional vectors are projected into an r dimensional subspace using a linear transformation. The transformation and its inverse for reconstruction is given by Haralick to be:

$$Y_i^j = V_i^T X_j \qquad i = 1, 2, \ldots r$$

where V_i, i = 1, 2, ... r are the basis vectors

$$\hat{X}_j = \sum_{k=1}^{r} \hat{Y}_k^j \, V_k$$

where $\hat{X}_j$; is the reconstructed value of X_j, and $\hat{Y}_k^j$ is the quantitized value of the coefficients.

The algorithm for generating the basis vectors is given by Haralick et al. [9]. This transform has a fast implementation as well. This is accomplished by treating the two-dimensional subimage as a one-dimensional n^2 data vector and the basis set as the direct product of two lower spaces, i.e., a 16 x 16 subimage is viewed as a 256 x 1 data vector and transformed by the direct product of two (16 x 16) basis sets. The first operates in the rows and the second operates in the columns.

The Slant Transform:

In 1972, The Proceedings of IEEE, in a Symposium on Walsh Functions published an extensive collection of papers on image applications. A highly publicized application by Enomoto and Shibatu [10] of Japan had been extended to a general fast transform [11]. This orthogonal, "Slant Transform", basis is a discrete sawtooth waveform decreasing in uniform steps over its length. The advantage is that this type of basis utilizes the gradual brightness changes in an image line. The general decomposition form of this matrix is:

$$[S_N] = \frac{1}{N} \begin{bmatrix} 1 & 0 & & & 1 & 0 & & \\ a_N & b_N & & & -a_N & b_N & & \\ & \ddots & & & & \ddots & & \\ & & I_{\frac{N}{2}-2} & & & & I_{\frac{N}{2}-2} & \\ 0 & 1 & & & 0 & -1 & & \\ -b_N & a_N & & & b_N & a_N & & \\ & \ddots & & & & \ddots & & \\ & & I_{\frac{N}{2}-2} & & & & -I_{\frac{N}{2}-2} & \end{bmatrix} \begin{bmatrix} S_{\frac{N}{2}} & \\ & S_{\frac{N}{2}} \end{bmatrix}$$

where $a_1 = 1$

$b_{2N} = 1/\ 1+4a^2N$

$a_{2N} = 2b_{2N}\ a_N$

The Discrete Cosine:

In more recent work in development of fast transforms for unique compression, Ahmed et al. [12] defined the Discrete Cosine and demonstrated how it could be implemented using the fast Fourier Transform. This work, developed the Cosine basis set as a class of Chebyshev polynomials. The basis is given by letting m=0,1,...,N-1:

$$\left\{ \frac{1}{\sqrt{2}} , \cos \frac{(2m+1)k\pi}{16} , k=1,2,...N \right\}$$

where N is the size of the subimage. Ahmed compares these results with the eigenvectors of a correlation matrix with first order Markov properties. The implication using this basis is that it closely approximates the Karhunen-Loeve. Shanmugam [13] pointed out that this is not surprising since the Karhunen-Loeve basis vectors and Cosine basis are asymptotically equivalent as dimensionality gets large. Haralick [14] has since introduced a storage efficient way to implement the Discrete Cosine.

Singular Value Decomposition:

Andrews et al. [23] has suggested that Outer Product Expansion methods may be applied to bandwidth reduction. It is suggested that the concept of eigenvalue map, condition number, and rank of an image are then useful guides to computer storage and savings. Andrews investigates the expansion or decomposition of the image by considering it a matrix, G, which is an n^2 matrix and digitized such that the i^{th} row and j^{th} column correspond to the x and y spatial coordinates of a scene. Now any matrix (in this case G) may

be represented by:

$$G = UDV^t$$

where U and V are arbitrary unitary matrices and D is a matrix comprised of the coefficients of expansion of G.

The elements of D are:

$$d_{ij} = \bar{u}_i^t \; G \; \bar{v}_j$$

$\bar{u}_i$ and $\bar{v}_i$ being the columns of U, V respectively.

Now if D can be diagonalized with suitable matrices U and V the expansion we are left with is known as a singular value decomposition. Given that D will have r diagonal values the matrix G can be represented by:

$$G = \sum_{i=1}^{r} d_i \bar{u}_i \bar{v}_i^t$$

The d_i terms are known as singular values of G and $\bar{u}_i$, $\bar{v}_i$ are the corresponding singular vectors. To choose the appropriate values for U and V

$$GG^t = U\Lambda U^t$$

and

$$G^tG = V\Lambda V^t$$

where Λ is the diagonal eigenvalues of GG^t.

Then d_i (singular values) will be the square roots of the eigenvalues of G^tG. By ordering the eigenvalues in descending order the smallest mean square error is achieved.

The eigenvalue map is the plot of singular values as a function ordered index r. Andrews then defines the condition number as:

$$C(G) = \frac{d_{max}}{d_{min}} = \frac{d_1}{d_r}$$

The compression relations now becomes apparent. For the case of retaining k outer products we approximate the image by:

$$G_k = \sum_{i=1}^{k} d_i \bar{u}_i \bar{v}_i^t$$

and the L_2 distance norm is

$$||G-G_k||^2 = \sum_{i=k+1}^{r} d_i^2$$

where the image was defined as rank r.

This approach to compression has very many interesting possibilities. As an example the first and second eigenimages contain high and low frequencies. This is not available in the standard transform approach.

The generation of basis sets and two-dimensional transforms is not the complete story in image compression. Once an image has been transformed it must be coded for the desired compression and transmission channel. Numerous papers have been produced on this facet of the compression application. The most noteworthy are those by Max, [15], Hibibi [16] and Wintz and Tasto [17]. These approaches investigate the means of quantizing coefficients of the transformed image and the effects of simulated channel errors. In very general terms they may be classed into three categories: fixed, variable word length and adaptive. Fixed infers that the same number of bits will be used for coding each coefficient of each subimage. Variable length codes assigns a different number of bits to each coefficient of any one subimage, but the same coefficient of two different subimages will be coded identically. These schemes are adaptive from image to image. The adaptive systems assigns codes in an "instantaneous" manner. This affords coding changes in a line to line basis [18]. This presumes a method of segmenting the image into classes or areas. The coding used in this research is an adaption of the variable length codes. A fixed number of bits is assigned to each subimage and the distribution is dependent on the variance of the coefficients (see Chapter 3).

All of these transform methods have been utilized with single frame, monochromatic film. It is a natural extension and a necessary one to consider frame-to-frame problems. A collection of papers in the inter-frame subject can be found in Huang and Tretiak [19]. The interframe redundancy reduction is possible because of the high correlation between adjacent frames. Since the background information does not change significantly only a few picture elements are contributing new information. The transforms discussed previously are being applied to this problem by the University of Southern California in three dimensions [20]. Estimates of a 5:1 reduction in the number of bits needed for reconstruction are being made.

The transform domain coefficients are also being combined with Differential Pulse Code Modulation (DPCM). This approach is not entirely new. Haralick [21] and Habibi [22] applied DPCM internal to an image (intra-frame) to achieve an additional compression of 1.5 to 2.0 bits/picture element. The interframe DPCM presumes a means of storing at least one frame and the image difference should require less bit allocations for coding. This is generally true if the scenes are not flat (same average gray level). In these methods a threshold of change is decided upon (not necessarily defined) and a change is noticed if above the threshold. This new information is then transmitted.

A disadvantage of the three-dimensional transforms is storage requirements, while the DPCM is relatively simple and requires only one frame be stored. The conditional change (threshold) methods generate codes at uneven rates depending on type of motion requiring the necessary buffers which will limit the motion toleration.

The references on transforms, coding, and the interframe problem although numerous have not solved the problem of optimizing image transmission. They have, however, sparked

the interest of many researchers, each contributing his own unique insight and, therefore, creating a better understanding of system possibilities. Since this research deals with transform image compression, those research articles applied in this report have been emphasized.

3. Definition of the Objective

It is the objective of this study to evaluate transform bandwidth compression techniques for removing the spatial redundancy of typical photography and television sensors used on the Remotely Piloted Vehicle Systems. This study should further establish the feasibility of a system compression of 50:1. The goal of 50:1 compression ratio will be achieved by three stages:

1. Transform compression of 12:1
2. Differential Pulse Code Modulation (DPCM) to yield a compression of 1.5:1
3. Frame rate reduction of 3:1

In evaluating the possible transform compression schemes, the "best" transform will be utilized in the hybrid (transform/DPCM) approach. Specific transforms under analysis are:

1. Hadamard
2. Fourier
3. SLANT
4. DLB
5. Discrete Cosine
6. Fast Karhunen-Loeve
7. Principal Components

4. Image Data Compression

In the search for efficient transform and coding schemes orthogonal transformations with their decomposition properties, and matrix theory provide powerful tools within the digital environment. In finite dimensional vector space orthogonal transformations may be viewed as linear operators from one space to another which preserve inner products. By the analysis

of the eigenvalues and corresponding eigenvectors an optimum solution may be found in the mean squared error sense when based on the second order statistics of the image. Other orthogonal transformations may then be compared to the optimum and may be sufficiently accurate and considerably easier to implement. In this study, the optimum solution based on eigenvalues and eigenvectors has been considered as well as several other leading orthogonal transformation schemes. These approaches and their comparison to the optimum solution is considered in Chapter 4. But first, we must understand how compression of an image is accomplished. Transform image data compression is accomplished by squeezing the image through a small dimensional hole. The squeezed image is the compressed image and what results after having passed the image through the small dimensional hole is the reconstructed image. More precisely, given an M x M digital image obtained by an analog to digital conversion process, the image is considered to be divided into a set of non-overlapping subimages (or windows) of size n x n; $n<M$. If each n x n subimage is interpreted as a vector in an n^2 dimensional space, where each of the n^2 coordinates correspond to one of the n^2 picture elements (PELS) in the subimages, then the digital image can be represented as a collection of K vectors each of dimension n^2; $K = (M/n)^2$. (See Figure 1.1)

Now let N denote the dimension of each data vector, $(N=n^2)$, and $x_1, x_2, \ldots, x_k$ be the collection of all data vectors from the K non-overlapping windows in the subimage. The transform compression is achieved by the projection of these N-tuples or N-dimensional data vectors onto an R dimensional subspace using a set of R, N-dimensional basis vectors $v_1, v_2, \ldots, v_R$. It is obvious that in order to compress R must be less than N.

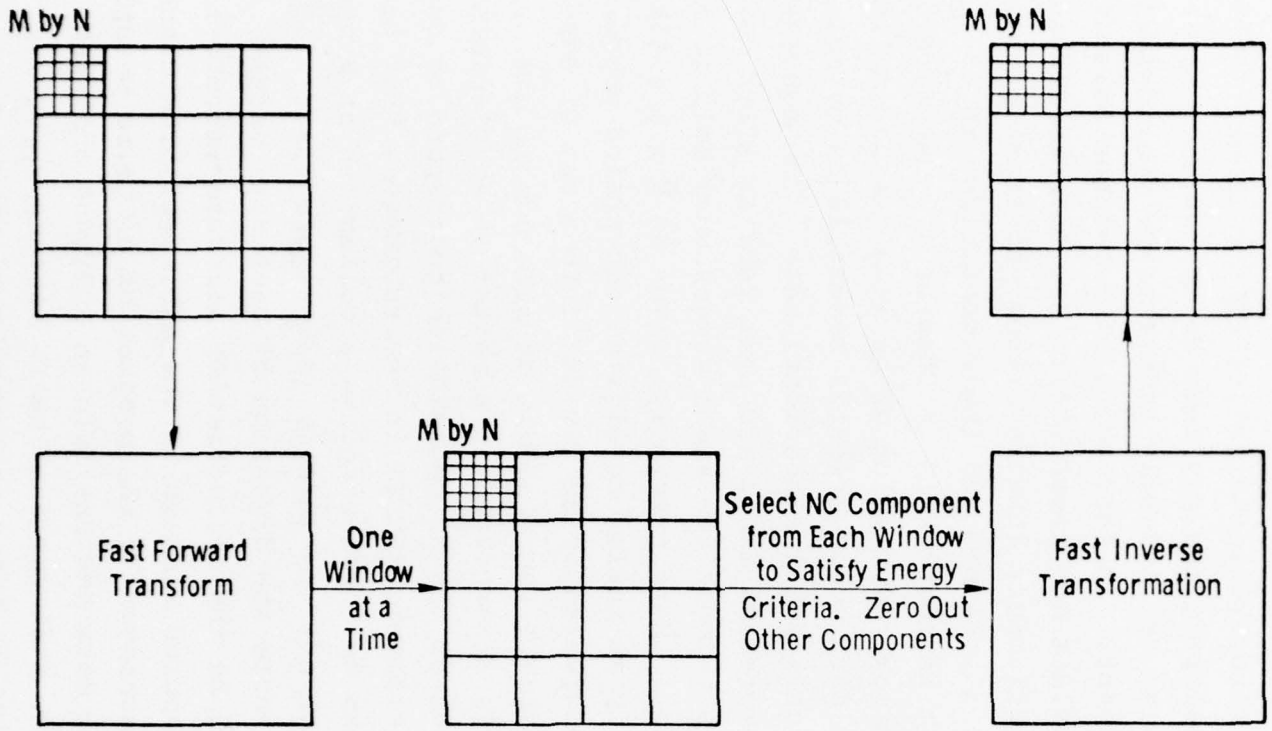


Figure 1.1. The transform coding technique

The i^{th} projection of the data vector x_j is given by $y_{ij}=v_i' x_j$ where $v_i' =(v_{i1}, v_{i2},\ldots,v_{iN})$ is the transpose of the i^{th} basis vector. The R projections of each x_j are coded independently and transmitted over a binary channel to the receiver. At the receiver, the reconstruction takes place by decoding the values of y_{ij} and approximating x_j with $\hat{x}_j$ according to:

$$\hat{X}_j = \sum_{i=1}^{R} Y_{ij} V_i$$

or in matrix notation:

$$\begin{pmatrix} | \\ \hat{X}_j \\ | \end{pmatrix}_{NX1} = \begin{pmatrix} | & | & & | \\ V_1 & V_2 & \cdots & V_R \\ | & | & & | \end{pmatrix}_{NXR} \begin{pmatrix} | \\ C_j \\ | \end{pmatrix}_{RX1}$$

$$\begin{pmatrix} | \\ C_j \\ | \end{pmatrix}_{RX1} = \begin{pmatrix} -V_1- \\ -V_2- \\ \vdots \\ -V_R- \end{pmatrix}_{RXN} \begin{pmatrix} | \\ X_j \\ | \end{pmatrix}_{NX1} , \quad j=1,\ldots,K$$

where c_j is the vector of R coefficients for any j^{th} vector. Of course, sometime during the transmission the basis vectors must be transmitted or known beforehand. The total squared error of reconstruction is:

$$\varepsilon^2 = \sum_{n=1}^{N} ||\hat{X}_n - X_n||^2$$

5. The Nature of Fast Transforms

Given a set of orthogonal basis vectors a fast algorithm is based on being able to decompose this matrix of basis vectors into several matrices where each row operation is less than what is required by the direct transform. Consider for a moment a one-dimensional transform. Given a basis set we can transform the data vector x of dimension N to another N dimensional vector whose components are the coordinates of the vector x in terms of the given basis set.

Let $v_1, v_2, \ldots, v_n$ be the given basis set. v_i is a column vector.

Let $V = \begin{pmatrix} — & v_1' & — \\ — & v_2' & — \\ & \vdots & \\ — & v_n' & — \end{pmatrix}$ be a matrix whose rows are the given basis vectors. The transform c of the vector x is given by:

$$C = VX$$

When the basis set is orthonormal, the inverse transform is given by X=V'C. This type of transform done in a brute force manner would require n^2 operations for a n x n matrix V. The fast transform implementation permits the transform to be done in 2n log n operations.

A fast transform implementation exists whenever the matrix V can be represented in a special way as a product of, say K square matrices, each square matrix having only 2 non-zero elements per row. When represented in this manner, the matrix V, which has 2^k rows and 2^k columns, can multiply a vector and require only $2k \cdot 2^k$ operations. The square matrices are often represented in a flow diagram with each layer in the diagram representing the multiplication by one square matrix.

The fast algorithm is based on the principal that the basis vectors of R^n can be represented by taking the direct product of the basis vectors of two lower dimensions. Let R^n be formed by the direct product of R^p and R^q such that $n = pq$. Let $U = (u_1, u_2, \ldots, u_p)$ and $V = (v_1, v_2, \ldots, v_q)$ be two vectors of dimension p and q respectively. Then we can define a direct product vector of dimension pq by:

$$UV = (u_1v_1, u_1v_2, \ldots u_1v_q, u_2v_1, u_2v_2, \ldots u_2v_q, \ldots, u_pv_1, u_pv_2, \ldots u_pv_q)$$

Now if we have a set of vectors $\{U_1, U_2, \ldots, U_p\}$ which are orthogonal basis for R^p and a set of vectors $\{V_1, V_2, \ldots, V_q\}$ which are an orthogonal basis for R^q then we can define a set of product vectors $\{U_iV_j \mid i=1,2,\ldots,p,\ j=1,2,\ldots q\}$ where each product vector is defined as above. As an example of a simple transform, Figure 1.2 shows the representation used for the more general case. For the forward transform case; let HD(i), i=1,2,...,n denote the dimension of each orthogonal basis set:

$$N = \prod_{i=1}^{n} HD(i)$$

The whole transformation is divided into n layers, one for each orthogonal basis set. Figure 1.3 is a pictorial representation of such a scheme with n=2; and, two basis sets comprised of:

$$H_2 = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\} \quad \text{and} \quad D_4 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \\ -1 \\ -3 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -3 \\ 3 \\ -1 \end{pmatrix} \right\}$$

Now each layer will have N boxes, each box representing a basis vector of the corresponding basis.

If $Mul(j) = \prod_{i=1}^{j} HD(i)$ the j^{th} layer will be comprised of N/MUL(j) sets of boxes. Each set contains MUL(j-1) boxes

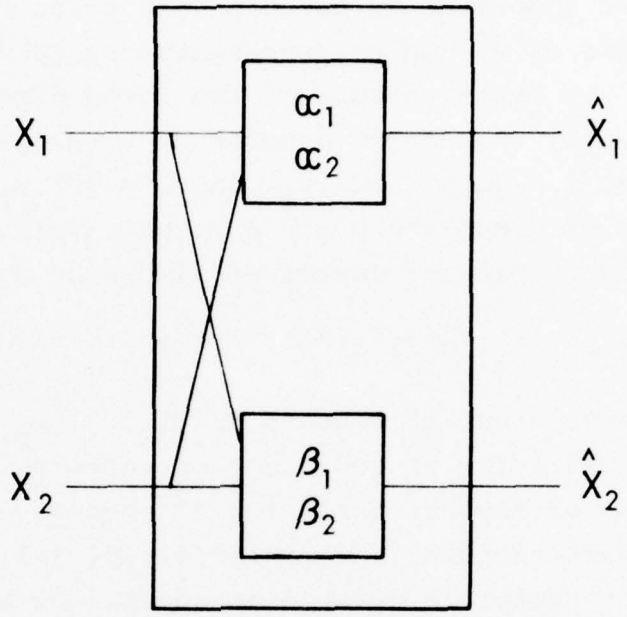


Illustration of a 2 X 2 transform defined by $\begin{pmatrix} \hat{X}_1 \\ \hat{X}_2 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$.

Thus, for example, the data point, X_1, leading to α_1, is multiplied by α_1 and the data point, X_2, leading to α_2 is multiplied by α_2. The results are then summed to produce $\hat{X}_1$.

Figure 1.2a.

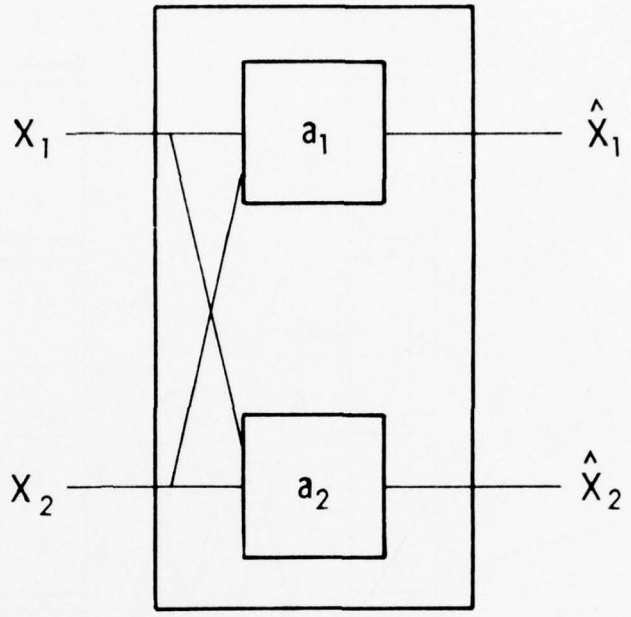


Illustration of a 2 X 2 transform defined by $\begin{pmatrix} \hat{X}_1 \\ \hat{X}_2 \end{pmatrix} = \begin{pmatrix} -a_1- \\ -a_2- \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$. We consider $a_1 = (\alpha_1 \alpha_2)$ and $a_2 = (\beta_1 \beta_2)$ to be row basis vectors. X_1 is then the projection of $X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$ on a_1 and X_2 is the projection of X on a_2.

Figure 1.2b.

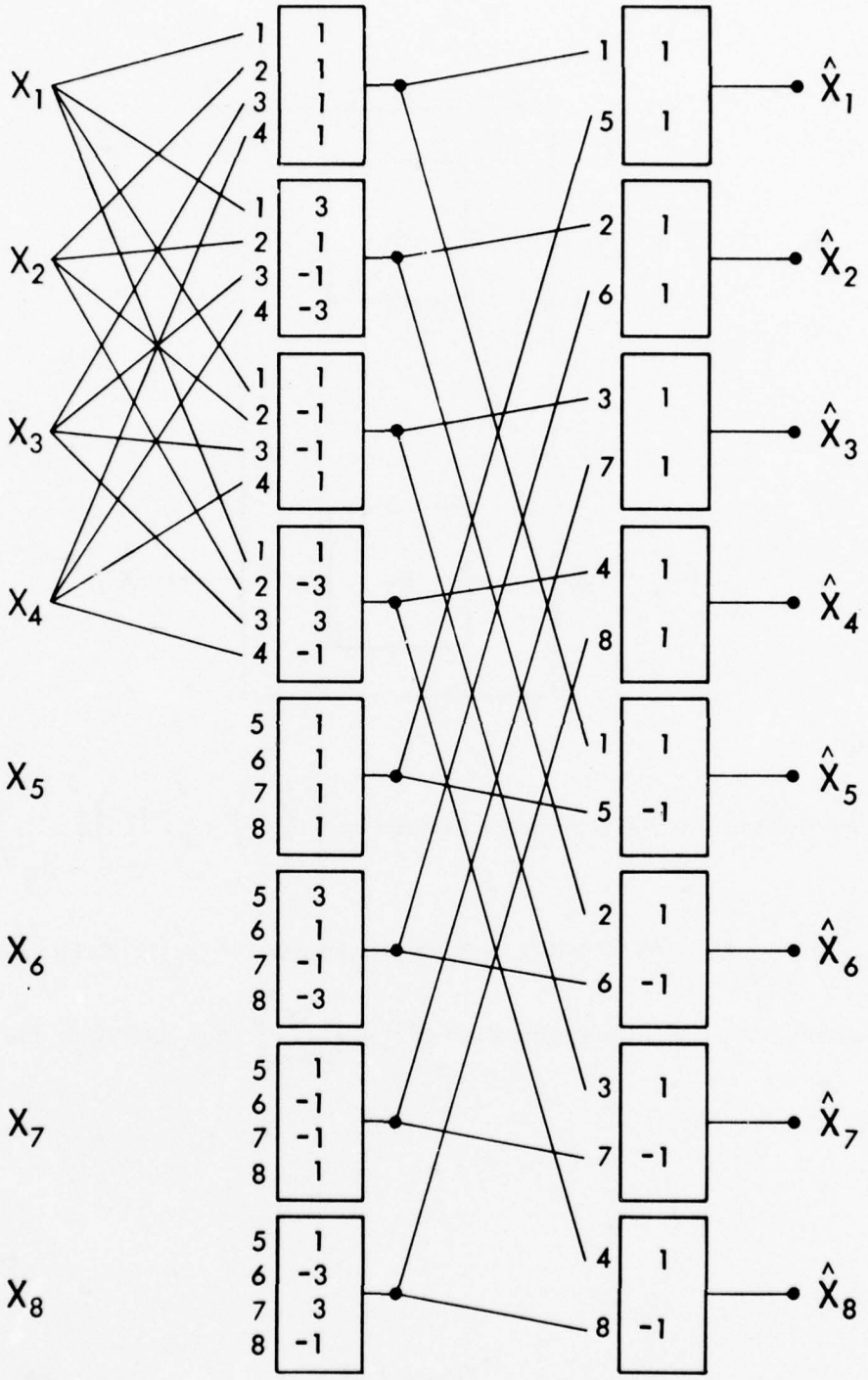


Figure 1.3. Layer Pattern for an 8 Dimensional Fast Transformation. N is Factored as 4·2.

all having the same basis vector repeated MUL(j-1) times in order. Each box in the j^{th} layer has HD(j) inputs with one output which is the dot product of the input components and the basis vector. If x(k), k=1,...,N is the input to the layer then the input to the b^{th} box is given as follows.

The starting index for each distinct basis vector of a set within the layer is the number of the first box in that set and is incremented by one for each repetition of that basis vector. Within each box the indexes of the input components are incremented by MUL(j-1) where M(o)=1. Haralick et al. [24] has shown that this scheme can be further simplified as in Figure 1.4. In this case each box in each layer is alike and represents all the vectors of the basis set. In this simplification each transform has an equal number of input and outputs of the same indexes. Each output is the dot product of the components input and the corresponding basis vector. There are further N/HD(i) transforms in the i^{th} layer. Haralick has defined the indexing of the input component as follows: "In the j^{th} layer the indexing of the input components for the k^{th} transform starts from (K-1)* HD(j) + 1 if MOD (K-1, MUL(j-1)) = 0, if not, the previous index is incremented by 1. Within the transform the indexes go by MUL(j-1)."[24] This forward transform is equivalent to obtaining the direct product of H_2 and D_4 given previously to form T_8 an 8 x 8 orthogonal basis set defined by $T_8=H_2 \times D_4 = \{t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8\}$ where

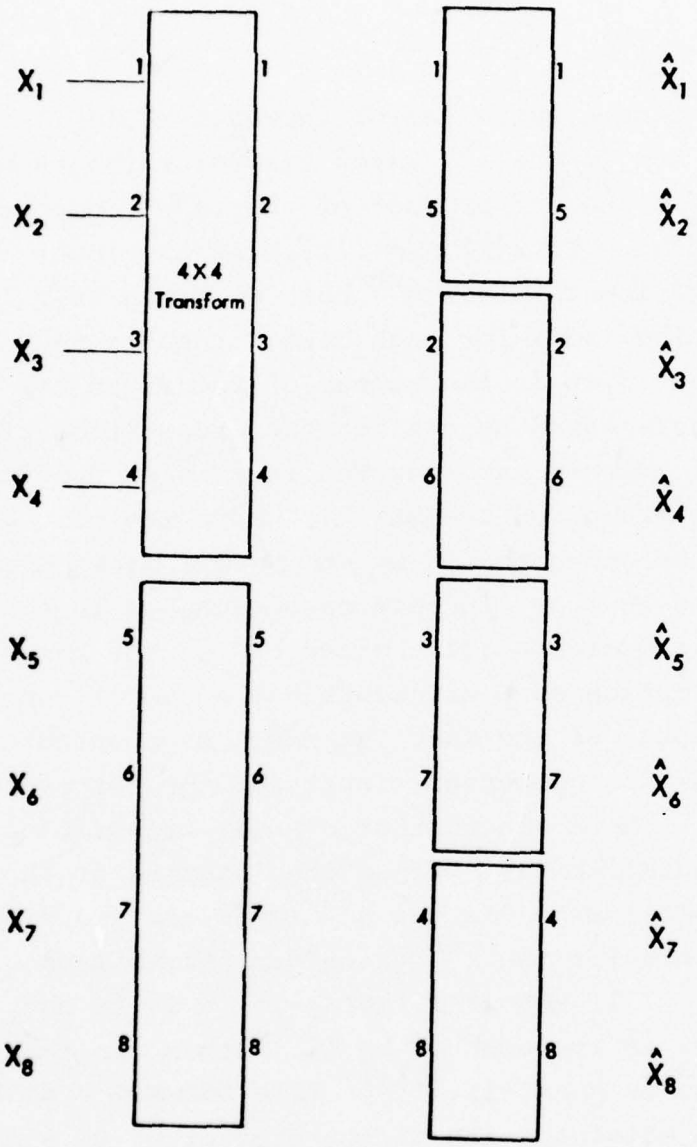


Figure 1.4. Illustration of the Indexing Scheme for a N=8 Fast Transform. In the First Layer There are Two 2(4x4) Transforms, in the Second Layer There are 4(2x2) for the Decomposition of N=8=4·2. The Numbers Labeling the Input Indicate What Position the Data Points are Coming From. The Numbers in the Output Side Label the Input Positions for the Next Layer.

$$t_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad t_2 = \begin{pmatrix} 3 \\ 1 \\ -1 \\ -3 \\ 3 \\ 1 \\ -1 \\ -3 \end{pmatrix} \quad t_3 = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} \quad t_4 = \begin{pmatrix} 1 \\ -3 \\ 3 \\ -1 \\ 1 \\ -3 \\ 3 \\ 1 \end{pmatrix}$$

$$t_5 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \quad t_6 = \begin{pmatrix} 3 \\ 1 \\ -1 \\ -3 \\ -3 \\ -1 \\ 1 \\ 3 \end{pmatrix} \quad t_7 = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} \quad t_8 = \begin{pmatrix} 1 \\ -3 \\ 3 \\ -1 \\ -1 \\ 3 \\ -3 \\ 1 \end{pmatrix}$$

Let T be a matrix whose rows are $t_1,\ldots,t_8$, respectively. T is the matrix transforming a vector into the coordinate system of T_8. This scheme can be applied to the inverse operation as well. Cleary $TT'=D$, where D is a diagonal matrix. Multiplying both sides by D^{-1}, $TT'D^{-1}=I$. Thus, the inverse of T is easily formed as $T^{-1}=T'D^{-1}$. Therefore if T is the square matrix of order n whose rows form the orthogonal basis set and $N_1,N_2,\ldots,N_n$ are the normed squares of the basis vectors, then the inverse definition is equal to:

$$T^{-1} = T'D^{-1}=T' \begin{bmatrix} \frac{1}{N_1} & & & \\ & \frac{1}{N_2} & & \\ & & \ddots & \\ & & & \frac{1}{N_n} \end{bmatrix}$$

The columns of the matrix T^{-1} will be the basis vectors for the co-responding layer in the inverse operation scheme. Figure 1.5 depicts the inverse layer operation which is the mirror image of the forward transform. The indexing scheme

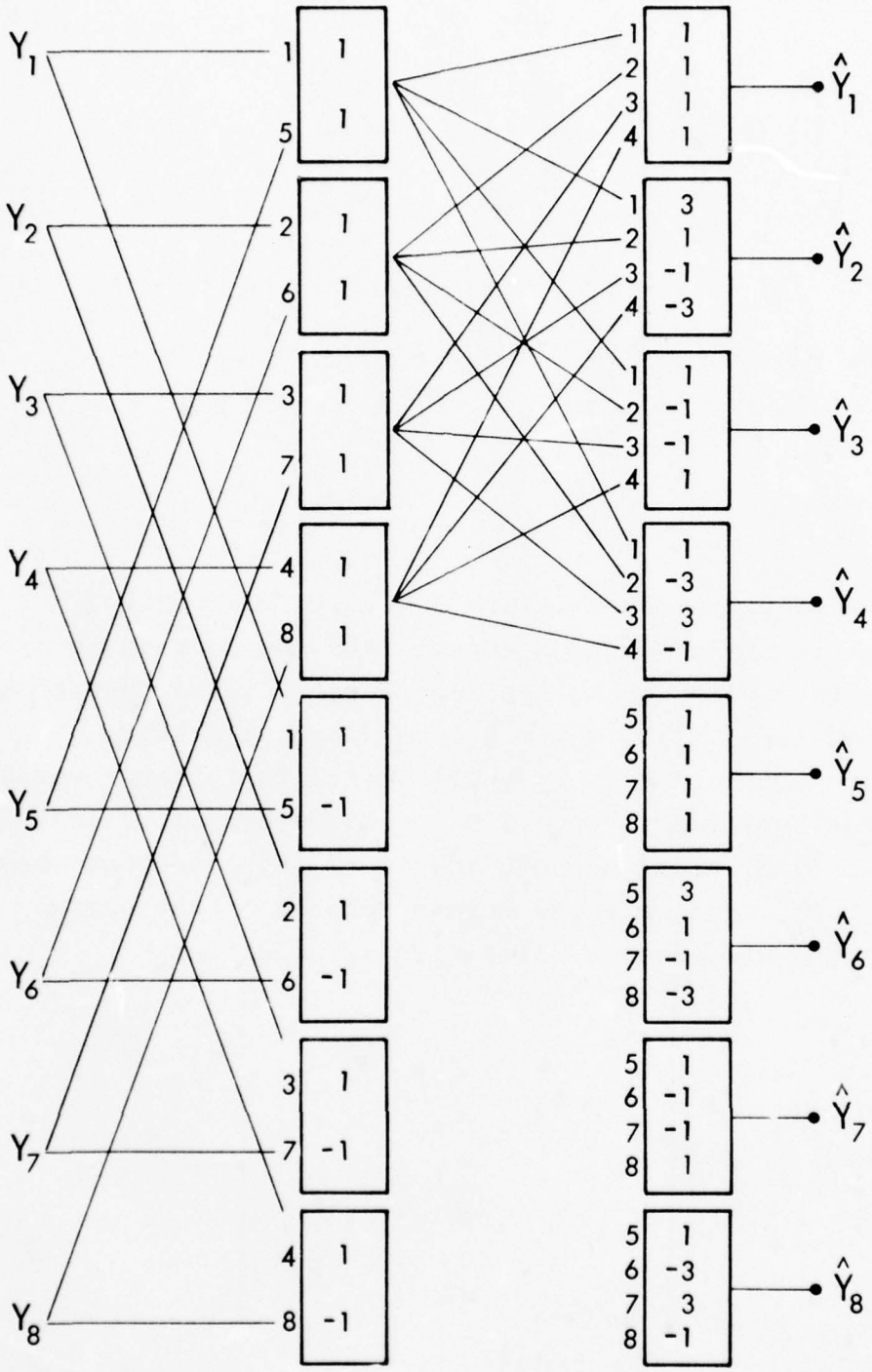


Figure 1.5. Layer Pattern for 8 Dimensional Fast Inverse Transformation. N is factored into $2 \cdot 4$.

is as follows: let HD(i)=1,...,n denoting the dimension of each inverse orthogonal basis.

$$N = \prod_{i=1}^{n} HD(i)$$

$$MULI\ (j) = \prod_{i=1}^{j} HD(i)$$

Then, also given by Haralick the indexing of "the input components for the j^{th} layer and k^{th} transform starts from (K-1)* HD(j)+1 if MOD((K-1), MULI(j+1))=0. If not, the previous starting index is incremented by 1. Within the transform the indexing goes by MULD (j+1)."[24]

6. The Error Criteria

The basic problem of transform coding is to determine what basis vectors are to be used in the projection operation. Crucial in such a choice is our criteria of best. In this particular program several indexes of performance have been used for evaluation of the imagery. These measures of performance are: 1) visual comparison of reconstructed images, 2) (RMS) the root-mean-square error between the original and reconstructed, 3) correlated RMS error.

In visual comparison the cosmetic qualities of the reconstructed image is compared to the original. Those major effects of the reconstructed image to be looked for are blocking, loss in resolution and contouring. Blockings is caused by reconstructing the image with an incomplete set of frequency components. Some of the components were set to zero in the coding process. If a poor choice of components has been used this will show up as discontinuities in gray level at the boundaries of the subimage used in the partitioning of the original image. This will also happen if insufficient bits are used in the coding scheme for

reconstruction. Loss in resolution is a spreading of boundaries noticable generally around man-made geometric objects or a fuzzy image. Contouring is the appearance of boundaries within the image rather than gradual gray level changes. This is generally caused by insufficient bits to reproduce the dynamic range of the gray levels within an image.

The RMS error has been accepted as a quality value in judging the reconstructed images. Although a poor criteria in general with regard to images it does provide a figure of merit of the errors which have accumulated in the transform-coding-reconstruction process. The root-mean-square error criterion may be stated as follows:

$$\text{RMS Error} = \sqrt{\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1} [I(i,j)-K(i,j)]^2}$$

where $I(i,j)$ represent elements of the original image and $K(i,j)$ represent elements of the reconstructed image. In this study images with $(256)^2$ elements and $(512)^2$ elements were used. The false impression of RMS error can be seen if an image has been enhanced in any manner during the process. The RMS error will be higher but the image may be more pleasing to the observer. This criteria must therefore be used in combination with other criteria. In images where no enhancement or non-linear process has been involved a large RMS error generally appears with poorer images.

Combining these two measures of fidelity with a measure of spatial correlation of errors yields a more complete idea of performance of transform compression. The correlated RMS error is formed in the following manner.

The gray values of the reconstructed images are rounded off to integers, the error image between original and reconstructed is obtained, and the RMS error computed from the error picture. The correlated RMS error is the product of

the RMS error times a spatial correlation measure obtained from the error image. This correlation measure, is a measure of spatial neighborly dependence of the error within the image. The computation is as follows:

Let $Z_x = \{1,\ldots,N_x\}$ and

$Z_y = \{1,\ldots,N_y\}$ be the set of x and y spatial domains of the error picture.

(row and columns indexes respectively)

Let $S = \{s_1, s_2 \ldots s_K\}$ be the set of gray tones in the error image. We first compute the relative frequency of occurrence of two neighboring cells separated by a distance whose radius is r, one with gray tone s_i and the other with gray tone value s_j.

$$I: \; Z_x x Z_y \rightarrow S$$

Let R be a binary relation defining the specified spatial relationship of any two resolution cells.

$$R \supseteq (Z_x x Z_y) \times (Z_x x Z_y) \text{ defined by}$$

$$R = \{((a,b),(c,d)) \mid \rho((a,b),(c,d)) = r\}$$

where ρ is a metric on $Z_x x Z_y$.

The joint probability density function may then be defined by:

$$P(S_i \; S_j) = \frac{\#\{((a,b),(c,d)) \in R \mid I(a,b) - S_i, I(c,d) = S_j\}}{\#R}$$

# denotes the number of elements in the set.

The marginal probability is given by:

$$P(S_i) = \frac{\#\{(n,m) \in (Z_x x Z_y) \mid I(n,m) = S_i\}}{\#(Z_x x Z_y)}$$

Now if we let $P(S_i, S_j)$ denote the normalized gray tone transition matrices then the correlation measure is defined by

the following equations: The average mutual information for two dimensions is:

$$C = \sum_{S_i} \sum_{S_j} P(S_i,S_j) \log_2 \frac{P(S_i,S_j)}{P(S_i)P(S_j)}$$

The source entropy is

$$h = \sum_{K} P(K) \log_2 P(K)$$

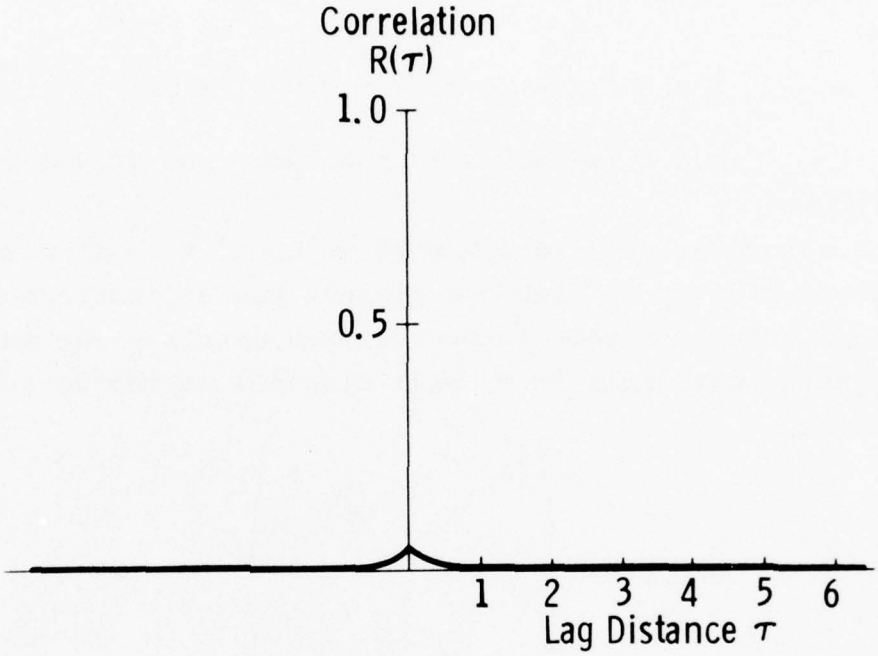
$$P(K) = \sum_{S_i} P(S_i,K) = \sum_{S_i} P(K,S_i)$$

and $$P = \frac{C}{h}$$

The average correlation measure contains information about the nature of the spatial distribution of the error picture and the RMS give a measure of the magnitude of the error between original and reconstructed. As an example a random Gaussian noise error picture was generated. The correlation function was generated with lag 1,2,5 (distance of cell separation) and the correlation was Ø as anticipated. See Figure 1.5.

7. Principal Components - An Optimum Transform in Terms of RMS Error Criteria

In view of the fact that all transform compression schemes should be compared to the optimum, and we have developed a two-dimensional fast Karhunen-Loeve (principal components) as part of this study, the principal components approach should be included as an integral part of background information necessary to the understanding of compression techniques. The K-L transform is optimum in the sense of minimum mean square error. In applying the K-L transform for data compression, the M x M original image is split up into K non-overlapping subimages of size n x n, where n<M,



Lag Dist.	R(τ)
1	1.03×10^{-5}
2	2.06×10^{-6}
3	1.67×10^{-6}

Figure 1.5. Correlation function, Gaussian random noise image.

windows. The elements of the n x n subimage form the n^2 elements of the vectors or windows. To do a K-L transform, the first step is to form the autocovariance which is defined by:

$$\Sigma = E\{(x-m)(x-m)'\} = E\{xx'\} - mm'$$

$m = E\{x\}$ and x is the vector of gray levels in a window.

The autocovariance matrix will be an $n^2 \times n^2$ matrix. The second step is to find the eigenvalues and corresponding eigenvectors of this autocovariance matrix. The matrix of eigenvectors, call it T, will diaganolize the covariance matrix.

$$T'\Sigma T = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$$

Let V be the matrix whose R rows are the eigenvectors with the M largest eigenvalues, of the $n^2 \times n^2$ covariance matrix. The K-L transform of a vector x is defined by $Y = V(x-m)$. Compression is achieved by the fact that $R<n$. The inverse transformation is defined by $x = v'y+m$. The total squared error of this process is the sum of the (m-n) smallest eigenvalues. Now by transforming the original random vector to the vector of principal components amounts to rotating of the coordinate system to a new system with inherent statistical properties. In terms of variance the first principal components is the normalized linear combinations of the original random variable with maximum variance, the second is the normalized linear combination having a variance greater than all others except the first and so on. The

coefficients generated by this method are not only uncorrelated but statistically independent. This is of particular importance in the image processing scheme since the coefficients energy contribution is based on the variance of the coefficient for a zero mean image. In fact, the orthogonal transformation using the eigenvectors leaves invariant the generalized variance and the sum of the variances of the components. This theorem is proved in Anderson [5] and repeated here as support for the above claim.

Theorem 1 - An orthogonal transformation of a random vector x leaves invariant the generalized variance and the sum of the variances of the components.

Let V be the transformation matrix whose R rows are the eigenvectors of the covariance matrix of dim. n., i.e., $V: R^n \to R^n$

$$C \text{ (vector of components)} = VX$$

$$E\{x\} = 0 \text{ and } E\{xx'\} = \Sigma$$

then $E\{c\} = \emptyset$ and $E\{cc'\} = vv'$

The generalized variance of C is:

$$|V\Sigma V'| = |V| \cdot |\Sigma| \cdot |V'| = |\Sigma||\, V'V| = |\Sigma|$$

which is the generalized variance of x.
The sum of the variances of the components of C is

$$\Sigma E\{C_i\}^2 = \text{trace}(V\Sigma V') = tr(\Sigma V'V) = tr(\Sigma I)$$
$$= \Sigma E\{x_i^2\}$$

In bandwidth compression (dimensionality reduction) we seek a function $f: R^n \to R^m$ where R^m is a smaller subspace of R^n such that the total squared error (ε^2) is minimized.

$$\min_{f} \sum_{k=1}^{K} (X_k - f(X_k))' \, (X_k - f(X_k))$$

Principal components states this minimum is achieved with an orthogonal projection operator projecting into the subspace spanned by the M largest eigenvectors corresponding to the M largest eigenvalues of the second moment matrix. To reiterate the total squared error is given by the (M-N) smallest eigenvalues.

Since this procedure yields the minimum error in terms of RMS criteria our investigation of orthogonal operators for bandwidth compression would be enhanced if a faster computational process approaching the minimum error could be found. In fact, this is possible, and discussed in the following chapter prior to discussing our results.

SECTION II

DEVELOPMENT OF A FAST TWO-DIMENSIONAL KARHUNEN-LOEVE TRANSFORM

1. INTRODUCTION

The purpose of transform coding is to store or represent data in a reduced dimensional space and yet preserve the data structure. When mean square error is the optimality criterion, the principal components or Karhunen-Loeve expansion is the best. However, the principal components technique is generally not used in transform coding image compression work because of its computational complexity. In this document we describe how to implement the principal components or Karhunen-Loeve transform (KLT) as a fast transform for image data compression when the image data satisfies mild stationarity and isotropic conditions.

Figure 2.1 illustrates the typical way transform coding is done. The N_r row and N_c columns image is partitioned into subimages or windows each having K_r rows and K_c columns. There are $(N_r/K_r) \cdot (N_c/K_c)$ such subimages. Each subimage is transformed using a Hadamard, Fourier, Slant, Discrete Cosine, or Discrete Linear Basis transform (all of which have fast implementations) or by the principal components or Karhunen-Loeve transform (which is slow). Those transform domain components which have the highest energy or variance are quantized and stored or transmitted while those transform domain components which have lowest energy or variance are not retained and effectively set to zero. Data compression is achieved because the number of bits required to encode the highest energy transform components are much less than the number of bits required to encode the data in its original spatial form.

In order to achieve the data compression by the principal components technique, the gray levels in each of the

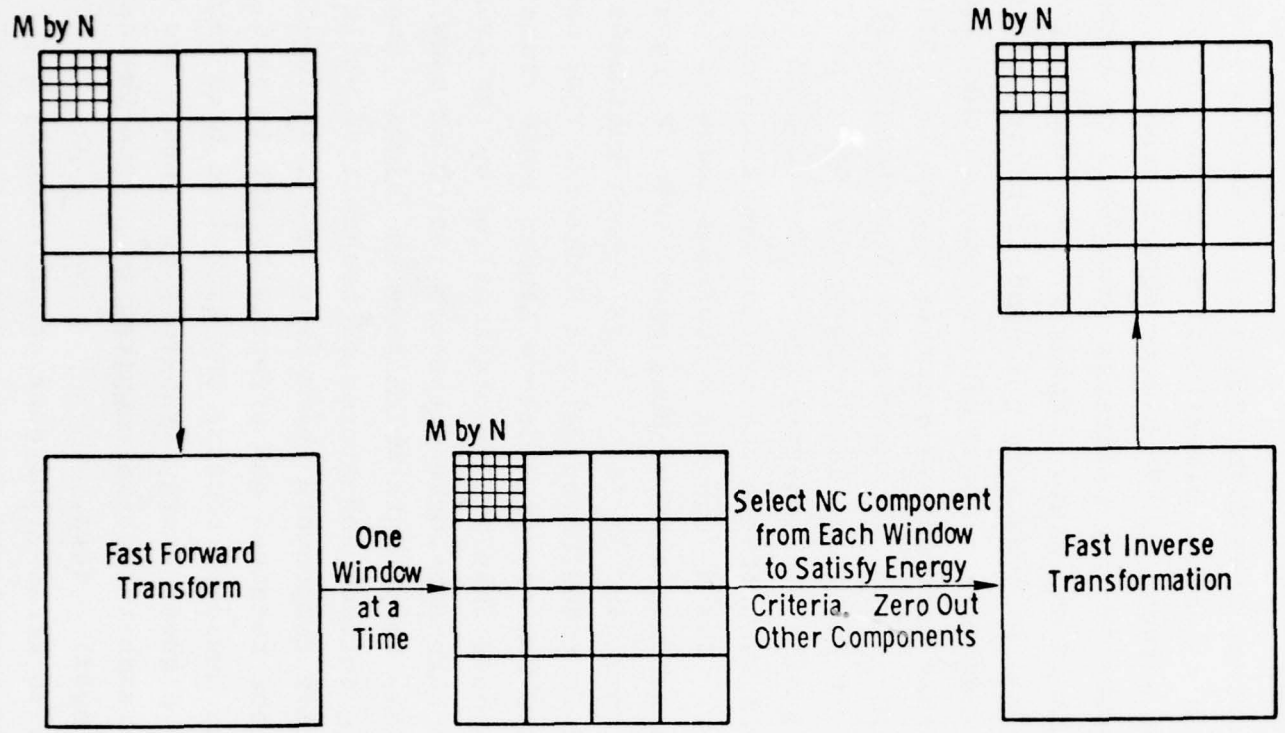


Figure 2.1. The transform coding technique

$(N_r/K_r) \cdot (N_c/K_c)$ subimages must be arranged as a vector and the $(K_rK_c) \times (K_rK_c)$ auto-covariance matrix for a sample fraction f of these vectors must be computed. This requires:

$$f\left(\frac{N_r}{K_r}\right)\left(\frac{N_c}{K_c}\right)(K_rK_c) = fN_rN_cK_rK_c$$

operations. Next the eigenvectors and eigenvalues of the auto-covariance matrix must be found. This requires on the order of $(K_rK_c)^3$ operations. To take the dot product of each subimage with K_rK_c vectors to obtain the transformed image requires $(N_r/K_r)\ (N_c/K_c)\ (K_rK_c)^2$ operations. The fast transform technique described here in will only require:

$$(N_r/K_r)\ (N_c/K_c)\ (K_r\ K_c)\ (K_r+K_c)$$

operations. This represents a savings factor of $K_rK_c/(K_r+K_c)$.

To begin our discussion of how a fast KLT may be developed, we must first discuss the general form of the auto-covariance and the kinds of matrices which may be diagonalized by fast transforms. This leads us to composite matrix theory. With this background, mild stationarity and isotropic assumptions will lead to the fast implementation by yielding a composite matrix which has a direct product form.

2. Forms of Matrices Which Fast Transforms Diagonalize

When dealing with the mean squared error criteria the optimal transform technique is the principal components, or by another name the Karhunen Loeve transform (KLT). Assuming the mean subimage has zero mean, the basis vectors of this transformation are given by the eigenvectors of the auto-covariance matrix for the set of subimages into which the image is partitioned. Thus, in order to do a good job of data compression we must select a fast transform technique whose basis vectors are the eigenvectors of matrices similar to the form of the auto-covariance matrix of the image.

To begin with, some examples (Figure 2.2) are presented which are alike in the sense that the eigenvectors depend only on the form of the matrix and not on any of the values of the elements of the matrix might take. Notice that in each of these examples, the eigenvalue, corresponding to an eigenvector is easily obtained as the dot product of the eigenvector with the first row of the matrix, e.g.,

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/3 & -1/3 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a + b + c + d \\ a + b/3 - c/3 - d \\ a - b - c + d \\ a - 3/b + 3c - d \end{pmatrix}$$

This occurs because the first components of each eigenvector is 1.

From this small set of 2nd, 3rd, and 4th order matrix eigenvector forms we can construct larger order matrix-eigenvector forms in a way consistent with the fast transform technique. Consider for example a composite matrix of the form

$$A = \begin{pmatrix} H_1 & H_2 & H_3 \\ H_2 & H_1-H_2+H_3 & H_2 \\ H_3 & H_2 & H_1 \end{pmatrix}$$

where H_1, H_2, and H_3 all have the same eigenvectors (they commute) and are of the form

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix}$$

The eigenvectors of A are of the form $x \cdot y$, the direct product of

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} a+b & 0 \\ 0 & a-b \end{pmatrix}$$

$$\begin{pmatrix} a & b & c \\ b & a-b+c & b \\ c & b & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -2 \\ 1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -2 \\ 1 & -1 & 1 \end{pmatrix}\begin{pmatrix} a+b+c & 0 & 0 \\ 0 & a-c & 0 \\ 0 & 0 & a-2b+c \end{pmatrix}$$

$$\begin{pmatrix} a & b & c & d \\ d & a & b & c \\ c & d & a & b \\ b & c & d & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+jb-c-jd \\ a-b+c-d \\ a-jb-c+jd \end{pmatrix}$$

$$\begin{pmatrix} a & b & c & d \\ b & a & d & c \\ c & d & a & b \\ d & c & b & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+d-c-d \\ a-b-c+d \\ a-b+c-d \end{pmatrix}$$

$$\begin{pmatrix} a & b & c & d \\ b & a-\frac{4b}{3}+\frac{4c}{3} & \frac{4b}{3}-\frac{4c}{3}+d & c \\ c & \frac{4b}{3}-\frac{4c}{3}+d & a-\frac{4b}{3}+\frac{4c}{3} & b \\ d & c & b & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/3 & -1 & -3 \\ 1 & -1/3 & -1 & 3 \\ 1 & -1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/3 & -1 & -3 \\ 1 & -1/3 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+\frac{b}{3}-\frac{c}{3}-d \\ a-b-c+d \\ a-3b+3c-d \end{pmatrix}$$

$$\begin{pmatrix} a & d & b & c \\ b & a & c & d \\ d & c & a & b \\ c & b & d & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & +j & -1 & -j \\ 1 & -1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & +j & -1 & -j \\ 1 & -1 & 1 & -1 \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+jb-c-jd \\ a-b+c-d \\ a-jd+c+jd \end{pmatrix}$$

$$\begin{pmatrix} a & c & d & b \\ c & a & b & d \\ b & d & a & c \\ d & b & c & a \end{pmatrix}\begin{pmatrix} 1 & 1 & +1 & 1 \\ 1 & -1 & +1 & -1 \\ 1 & -j & -1 & +j \\ 1 & j & -1 & -j \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & +j \\ 1 & j & -1 & -j \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+jb-c-jd \\ a-b+c-d \\ a-jb-c+jd \end{pmatrix}$$

$$\begin{pmatrix} q & r & s & t & u \\ r & q-r+\frac{t+s}{2} & \frac{r+t}{2} & u-t+\frac{s+r}{2} & t \\ s & \frac{r+t}{2} & q+u-s & \frac{r+t}{2} & s \\ t & u-t+\frac{s+r}{2} & \frac{r+t}{2} & q-r+\frac{t+s}{2} & r \\ u & t & s & r & q \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & +1 & 1 \\ 1 & 1/2 & 0 & -2 & -3/2 \\ 1 & 0 & -2 & 0 & 1 \\ 1 & -1/2 & 0 & -2 & -3/2 \\ 1 & -1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1/2 & 0 & -2 & -3/2 \\ 1 & 0 & -2 & 0 & 1 \\ 1 & -1/2 & 0 & -2 & -3/2 \\ 1 & -1 & 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} q+r+s+t+u \\ q+1/2r-1/2t-u \\ q-2s+u \\ q-2r-2t-u \\ q-\frac{3r}{2}+s-\frac{3t}{2}+u \end{pmatrix}$$

Figure 2.2. Examples of matrices in which eigenvectors depend only on form of matrix.

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

an eigenvector of a matrix of the form

$$\begin{pmatrix} a & b & c \\ b & a-b+c & b \\ c & b & a \end{pmatrix}$$

with

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

an eigenvector of a matrix of the form

$$\begin{pmatrix} d & e \\ e & d \end{pmatrix}$$

See Afriat, (1954)[1], and Williamson, (1931)[2]. Why this must be so is easily illustrated in our example. Consider

$$A(x \cdot y) = \begin{pmatrix} H_1 & H_2 & H_3 \\ H_2 & H_1 - H_2 + H_3 & H_2 \\ H_3 & H_2 & H_1 \end{pmatrix} \begin{pmatrix} x_1 y \\ x_2 y \\ x_3 y \end{pmatrix}$$

$$= \begin{pmatrix} x_1 H_1 y + x_2 H_2 y + x_3 H_3 y \\ x_1 H_2 y + x_2 (H_1 - H_2 + H_3) y + x_3 H_2 y \\ x_1 H_3 y + x_2 H_2 y + x_3 H_1 y \end{pmatrix}$$

$$= \begin{pmatrix} x_1 n_1 y + x_2 n_2 y + x_3 n_3 y \\ x_1 n_2 y + x_2 (n_1 y - n_2 y + n_3 y) + x_3 n_2 y \\ x_1 n_3 y + x_2 n_2 y + x_3 n_1 y \end{pmatrix}$$

$$= \begin{pmatrix} x_1 n_1 + x_2 n_2 + x_3 n_3 \\ x_1 n_2 + x_2 (n_1 - n_2 + n_3) + x_3 n_2 \\ x_1 n_3 + x_2 n_2 + x_3 n_1 \end{pmatrix} \cdot y$$

$$= \left[\begin{pmatrix} n_1 & n_2 & n_3 \\ n_2 & n_1 - n_2 + n_3 & n_2 \\ n_3 & n_2 & n_1 \end{pmatrix} x \right] \cdot y$$

But since x is an eigenvector of any matrix of the form

$$\begin{pmatrix} a & b & c \\ b & a-b+c & b \\ c & b & a \end{pmatrix} \quad , \quad \begin{pmatrix} n_1 & n_2 & n_3 \\ n_2 & n_1-n_2+n_3 & n_2 \\ n_3 & n_2 & n_1 \end{pmatrix} \quad x = \lambda x$$

Hence,

$$A(x \cdot y) = [\lambda x] \cdot y = (x \cdot y).$$

Thus, if y is an eigenvector of H_1, H_2, and H_3 so that $H_1 y = n_1 y$, $H_2 y = n_2 y$, and $H_3 y = n_3 y$ and x is an eigenvector of

$$B = \begin{pmatrix} n_1 & n_2 & n_3 \\ n_2 & n_1-n_2+n_3 & n_2 \\ n_3 & n_2 & n_1 \end{pmatrix}$$

the matrix B having the same form as A except that the corresponding eigenvalues appear in place of the H matrices, and x has corresponding eigenvalue λ, then $x \cdot y$ is an eigenvector of A, with eigenvalue λ. In this example the eigenvectors of A are

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ -2 \\ -2 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ -2 \\ 2 \\ 1 \\ -1 \end{pmatrix}$$

and they correspond to the fast transform implemented as the direct product of the vectors in

$$\left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} \right\}$$

with those in

$$\left\{\begin{pmatrix}1\\1\end{pmatrix} \quad \begin{pmatrix}1\\-1\end{pmatrix}\right\}$$

The matrix A has the general form

$$\begin{bmatrix} a & b & c & d & e & f \\ b & a & d & c & f & e \\ c & d & (a-c+e) & (b-d+f) & c & d \\ d & c & (b-d+f) & (a-c+e) & d & c \\ e & f & c & d & a & b \\ f & e & d & c & b & a \end{bmatrix}$$

and has eigenvalues corresponding to the 6 listed eigenvectors given respectively by

$(a+b+c+d+e+f)$, $(a-b+c-d+e-f)$, $(a+b-c-f)$
$(a-b-e+f)$, $(a+b-2c-2d+e+f)$, and $(a-b-2c+2d+e-f)$.

3. Stationarity and Isotropicity

In general, however, our auto-covariance matrix will not have the previous discussed form without assumptions of stationarity and isotropicity. [3] Suppose an image I had N_r rows and N_c columns. Partition this image into mutually exclusive subimages each K_r rows by K_c columns. (We assume N_r is an interger multiple of K_r and N_c is an interger multiple of K_c.) We say the image I is stationary (in the weak sense) when two conditions are satisfied:

(1) The mean gray tone of all resolution cells situated in subimage row column coordinates (i,j) is the same constant μ which is independent of the relative subimage coordinates (i,j).

(2) The gray tone covariance of all pairs of resolution cells situated in subimage row column coordinates $[(i,j), (i+n_1, j+n_2)]$ is a function $\sigma(n_1 n_2)$ only of the row column translations $(n_1 n_2)$ and independent of the relative subimage coordinates (i,j).

As illustrated in Figure 2.3, condition (1) implies, for example, that the average gray tone of all resolution cells occupying the upper left hand corner of the subimages equals the average of all resolution cells occupying the upper right hand corner of the subimages. In other words, fix a relative coordinate of the subimage. Then the average gray tone taken over all resolution cells having those relative coordinates in the subimages is equal to the average taken over any other relative coordinates.

As illustrated in Figure 2.4, condition (2) implies, for example, that the average second moment gray tone taken between resolution cells occupying the first and second columns of the first row in each subimage must equal the average second moment gray tone taken between resolution cells occupying the fifth and sixth columns of the third row in each subimage. In other words, fix some relative coordinates of the subimage. Then choose a row column translation. Then the second moment gray tone taken between all resolution cells situated in the specified relative coordinates and in the specified relative coordinates shifted by the row and column translation is independent of the specified relative coordinates and only a function of the translation factor.

An image I is isotropic if it is stationary and if the covariance depends only on the spatial distance of the translations. Thus, for example, the covariance for a shift of one row down and two columns over is not necessarily equal to the covariance for a shift of two rows down and one column over for a stationary image, but since they both represent translation of distance $\sqrt{5}$ they would be equal for an isotropic image.

In more mathematical terms, let

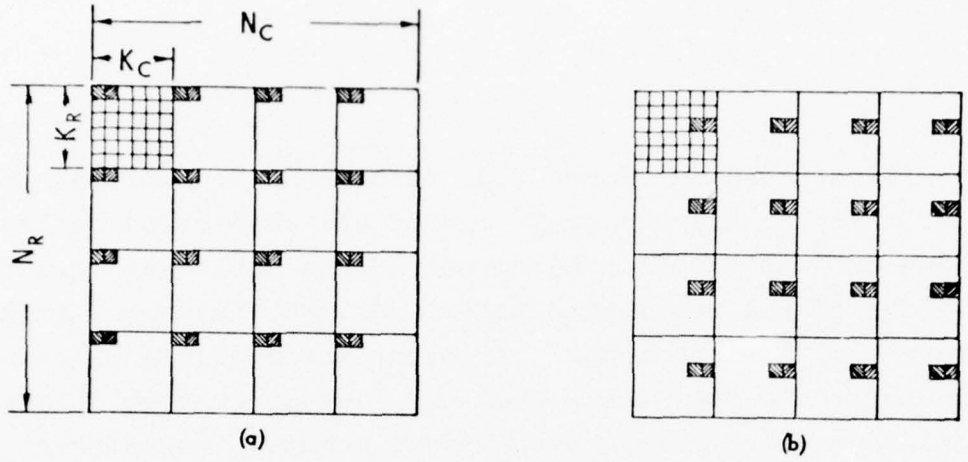


Figure 2.3. Illustrates that in a stationary image the average greytone taken over all resolution cells marked in image (a) equals as the average greytone taken over all resolution cells marked in image (b).

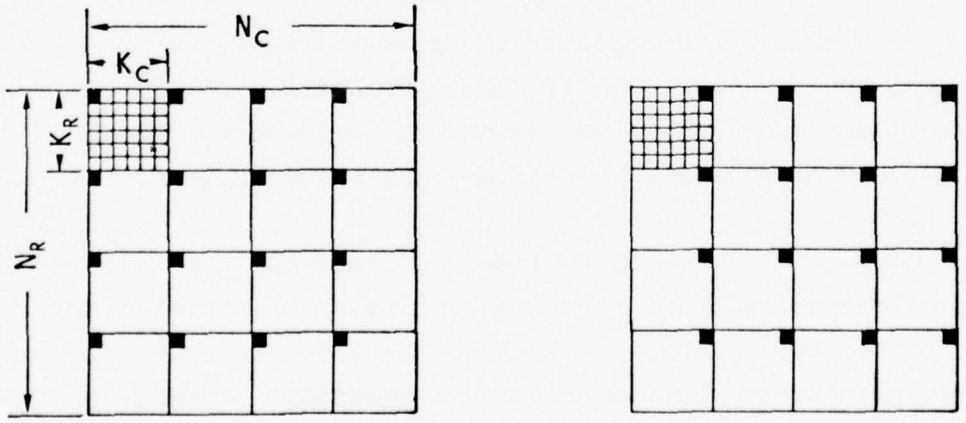


Figure 2.4. Illustrates that in stationary images the second moment statistics taken over all resolution cells marked in image (a) equals the second moment statistics taken over all resolution cells marked in image (b).

$Z_r = \{0,1,\ldots,N_r-1\}$ be the set of row indexes for digital image I

$Z_c = \{0,1,\ldots,N_c-1\}$ be the set of column indexes for digital image I.

Suppose N_r is an integer multiple of K_r and N_c is an integer multiple of K_c. Let

$$R(i,j) = \{(a,b)\varepsilon(Z_r \times Z_c) \mid a \text{ modulo } K_r = i \text{ and } b \text{ modulo } K_c = j\}$$

$$S(i,j,n_1,n_2) = \{[(a,b),(c,d)]\varepsilon(Z_r \times Z_c) \times (Z_r \times Z_c) \mid c = a + n_1,\ d = b + n_2, \text{ and } (a,b)\varepsilon R(i,j)\}$$

An image I: $Z_r \times Z_c \rightarrow G$ is stationary if and only if

$$(1)\quad \mu = \frac{1}{\#R(i,j)} \sum_{(a,b)\varepsilon R(i,j)} I(a,b) \quad \text{for each } (i,j)$$

$$(2)\quad \sigma(n_1,n_2) = \frac{1}{\#S(i,j,n_1,n_2)} \sum_{[(a,b),(c,d)]\varepsilon S(i,j,n_1,n_2)} (I(a,b)-\mu)(I,(c,d)-\mu)$$

for each (i,j)

An image I: $Z_r \times Z_c \rightarrow G$ is isotropic if and only if I is stationary and

$$\sigma(n_1n_2) = \sigma(m_1,m_2) \text{ whenever } n_1^2 + n_2^2 = m_1^2 + m_2^2.$$

Consider now how we could compute the covariance matrix for an image which satisfies or almost satisfies the stationarity or isotropicity conditions. Shown in Figure 2.5 is a 16 x 16 image. We will partition it into 4 x 4 subimages (Figure 2.6). Figure 2.7 shows the general form of the autocovariance matrix when no stationarity or isotropicity assumptions are made. To determine the covariance matrix with stationarity assumed, we generate a tableau which depicts all pairs of resolution cells situated in the same translational

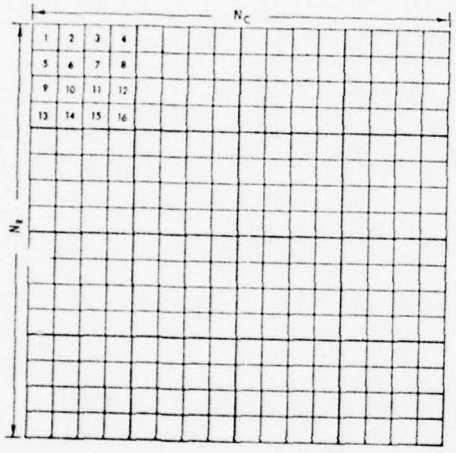


Figure 2.5. Illustrates a 16 x 16 image partitioned into 4 x 4 subimages.

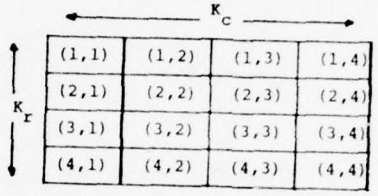


Figure 2.6. A K_r x K_c subimage of the original image.

Resolution Cell	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)	(4,1)	(4,2)	(4,3)	(4,4)
(1,1)	aa	ab	ac	ad	ae	af	ag	ah	ai	aj	ak	al	am	an	ao	ap
(1,2)		bb	bc	bd	be	bf	bg	bh	bi	bj	bk	bl	bm	bn	bo	bp
(1,3)			cc	cd	ce	cf	cg	ch	ci	cj	ck	cl	cm	cn	co	cp
(1,4)				dd	de	df	dg	dh	di	dj	dk	dl	dm	dn	do	dp
(2,1)					ee	ef	eg	eh	ei	ej	ek	el	em	en	eo	ep
(2,2)						ff	fg	fh	fi	fj	fk	fl	fm	fn	fo	fp
(2,3)							gg	gh	gi	gj	gk	gl	gm	gn	go	gp
(2,4)								hh	hi	hj	hk	hl	hm	hn	ho	hp
(3,1)									ii	ij	ik	il	im	in	io	ip
(3,2)										jj	jk	jl	jm	jn	jo	jp
(3,3)											kk	kl	km	kn	ko	kp
(3,4)												ll	lm	ln	lo	lp
(4,1)													mm	mn	mo	mp
(4,2)														nn	no	np
(4,3)															oo	op
(4,4)																pp

Figure 2.7. Illustrates that the auto-covariance matrix for a non-stationary image partitioned into 4 x 4 subimages, has 136 distinct entries. Each distinct entry is labeled with a two character label.

relationship [see Table 2.1]. From this table the stationary autocovariance may be formed, Figure 2.8. If the tableau is modified by the definition of isotropicity we can generate a new table and subsequent covariance array, Table 2.2, and Figure 2.9. The isotropic assumption basically establishes equivalence classes of the lettered columns in Table 2.1 in the following manner:

{a, g}	{x, r}
{q, s}	{t, ℓ, p, j}
{v, d}	{f, n, b, h}
{w, k}	{ə, u, c, o}
{m, i}	

4. Symmetry Properties of a Covariance Matrix of an Isotropic Image

A covariance matrix of an isotropic image has symmetry properties which permit the rapid computation of its eigenvectors. Such a covariance matrix can always be partitioned into K_r^2 (K_c x K_c) submatrices, each submatrix being of the same toeplitz form. Each such toeplitz submatrix has only K_c distinct entries as shown in Figure 2.10. The number of times each distinct entry occurs is given by:

Entry name	Number of times entry occurs
v_1	$n_1 = K_c$
v_2	$n_2 = 2n_1 - 2$
v_3	$n_3 = n_2 - 2$
v_4	$n_4 = n_3 - 2$
$\vdots$	
v_{K_c-1}	$n_{K_c-1} = n_{K_c-2} - 2$
v_{K_c}	$n_{K_c} = n_{K_c-1} - 2 = 2$

The total number of distinct submatrices in the partition of the covariance matrix is K_r. The submatrices themselves are organized as a toeplitz form. As shown in Figure 2.11 the number of times each submatrix is repeated in the covariance matrix is given by:

a	b	c	d	e	f	g
(3,3)	(3,2)	(3,1)	(3,0)	(3,-1)	(3,-2)	(3,-3)
(1,1)-(4,4)	(1,1)-(4,3) (1,2)-(4,4)	(1,1)-(4,2) (1,2)-(4,3) (1,3)-(4,4)	(1,1)-(4,1) (1,2)-(4,2) (1,3)-(4,3) (1,4)-(4,4)	(1,2)-(4,1) (1,3)-(4,2) (1,4)-(4,3)	(1,3)-(4,1) (1,4)-(4,2)	(1,4)-(4,1)

h	i	j	k	t	m	n
(2,3)	(2,2)	(2,1)	(2,0)	(2,-1)	(2,-2)	(2,-3)
(1,1)-(3,4) (2,1)-(4,4)	(1,1)-(3,3) (1,2)-(3,4) (2,1)-(4,3) (2,2)-(4,4)	(1,1)-(3,2) (1,2)-(3,3) (1,3)-(3,4) (2,1)-(4,2) (2,2)-(4,3) (2,3)-(4,4)	(1,1)-(3,1) (1,2)-(3,2) (1,3)-(3,3) (1,4)-(3,4) (2,1)-(4,1) (2,2)-(4,2) (2,3)-(4,3) (2,4)-(4,4)	(1,2)-(3,1) (1,3)-(3,2) (1,4)-(3,3) (2,2)-(4,1) (2,3)-(4,2) (2,4)-(4,3)	(1,3)-(3,1) (1,4)-(3,2) (2,3)-(4,1) (2,4)-(4,2)	(1,4)-(3,1) (2,4)-(4,1)

o	p	q	r		s	t
(1,3)	(1,2)	(1,1)	(1,0)		(1,-1)	(1,-2)
(1,1)-(2,4) (2,1)-(3,4) (3,1)-(4,4)	(1,1)-(2,3) (1,2)-(2,4) (2,1)-(3,3) (2,2)-(3,4) (3,1)-(4,3) (3,2)-(4,4)	(1,1)-(2,2) (1,2)-(2,3) (1,3)-(2,4) (2,1)-(3,2) (2,2)-(3,3) (2,3)-(3,4) (3,1)-(4,2) (3,2)-(4,3) (3,3)-(4,4)	(1,1)-(2,1) (1,2)-(2,2) (1,3)-(2,3) (1,4)-(2,4) (2,1)-(3,1) (2,2)-(3,2) (2,3)-(3,3) (2,4)-(3,4)	(3,1)-(4,1) (3,2)-(4,2) (3,3)-(4,3) (3,4)-(4,4)	(1,2)-(2,1) (1,3)-(2,2) (1,4)-(2,3) (2,2)-(3,1) (2,3)-(3,2) (2,4)-(3,3) (3,2)-(4,1) (3,3)-(4,2) (3,4)-(4,3)	(1,3)-(2,1) (1,4)-(2,2) (2,3)-(3,1) (2,4)-(3,2) (3,3)-(4,1) (3,4)-(4,2)

u	v	w	x		y	
(1,-3)	(0,3)	(0,2)	(0,1)		(0,0)	
(1,4)-(2,1) (2,4)-(3,1) (3,4)-(4,1)	(1,1)-(1,4) (2,1)-(2,4) (3,1)-(3,4) (4,1)-(4,4)	(1,1)-(1,3) (1,2)-(1,4) (2,1)-(2,3) (2,2)-(2,4) (3,1)-(3,3) (3,2)-(3,4) (4,1)-(4,3) (4,2)-(4,4)	(1,1)-(1,2) (1,2)-(1,3) (1,3)-(1,4) (2,1)-(2,2) (2,2)-(2,3) (2,3)-(2,4) (3,1)-(3,2) (3,2)-(3,3)	(3,3)-(3,4) (4,1)-(4,2) (4,2)-(4,3) (4,3)-(4,4)	(1,1)-(1,1) (1,2)-(1,2) (1,3)-(1,3) (1,4)-(1,4) (2,1)-(2,1) (2,2)-(2,2) (2,3)-(2,3) (2,4)-(2,4)	(3,1)-(3,1) (3,2)-(3,2) (3,3)-(3,3) (3,4)-(3,4) (4,1)-(4,1) (4,2)-(4,2) (4,3)-(4,3) (4,4)-(4,4)

Table 2.1. Lists in each column all pairs of resolution cells situated in the same translation relationship. For example, all the resolution cell pairs listed under column c are related by three rows down and one row across.

Resolution Cell	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)	(4,1)	(4,2)	(4,3)	(4,4)
(1,1)	y	x	w	v	r	q	p	o	k	j	i	h	d	c	b	a
(1,2)	x	y	x	w	s	r	q	p	l	k	j	i	e	d	c	b
(1,3)	w	x	y	x	t	s	r	q	m	l	k	j	f	e	d	c
(1,4)	v	w	x	y	u	t	s	r	n	m	l	k	g	f	e	d
(2,1)	r	s	t	u	y	x	w	v	r	q	p	o	k	j	i	h
(2,2)	q	r	s	t	x	y	x	w	s	r	q	p	l	k	j	i
(2,3)	p	q	r	s	w	x	y	x	t	s	r	q	m	l	k	j
(2,4)	o	p	q	r	v	w	x	y	u	t	s	r	n	m	l	k
(3,1)	k	l	m	n	r	s	t	u	y	x	w	v	r	q	p	o
(3,2)	j	k	l	m	q	r	s	t	x	y	x	w	s	r	q	p
(3,3)	i	j	k	l	p	q	r	s	w	x	y	x	t	s	r	q
(3,4)	h	i	j	k	o	p	q	r	v	w	x	y	u	t	s	r
(4,1)	d	e	f	g	k	l	m	n	r	s	t	u	y	x	w	v
(4,2)	c	d	e	f	j	k	l	m	q	r	s	t	x	y	x	w
(4,3)	b	c	d	e	i	j	k	l	p	q	r	s	w	x	y	x
(4,4)	a	b	c	d	h	i	j	k	o	p	q	r	v	w	x	y

Figure 2.8 Form of auto-covariance with stationary assumption.

a	b	c	d
$d^2=18$	$d^2=13$	$d^2=10$	$d^2=9$
(1,1)-(4,4)	(1,1)-(4,3)	(1,1)-(4,2)	(1,1)-(4,1)
(1,4)-(4,1)	(1,2)-(4,4)	(1,2)-(4,3)	(1,2)-(4,2)
	(1,3)-(4,1)	(1,3)-(4,4)	(1,3)-(4,3)
	(1,4)-(4,2)	(1,2)-(4,1)	(1,4)-(4,4)
	(1,4)-(3,1)	(1,3)-(4,2)	(1,1)-(1,4)
	(2,4)-(4,1)	(1,4)-(4,3)	(2,1)-(2,4)
	(1,1)-(3,4)	(1,4)-(2,1)	(3,1)-(3,4)
	(2,1)-(4,4)	(2,4)-(3,1)	(4,1)-(4,4)
		(3,4)-(4,1)	
		(1,1)-(2,4)	
		(2,1)-(3,4)	
		(3,1)-(4,4)	

f	j		k	
$d^2=8$	$d^2=5$		$d^2=4$	
(1,1)-(3,3)	(1,1)-(3,2)	(1,2)-(3,1)	(1,1)-(3,1)	(1,1)-(1,3)
(1,2)-(3,4)	(1,2)-(3,3)	(1,3)-(3,2)	(1,2)-(3,2)	(1,2)-(1,4)
(2,1)-(4,3)	(1,3)-(3,4)	(1,4)-(3,3)	(1,3)-(3,3)	(2,1)-(2,3)
(2,2)-(4,4)	(2,1)-(4,2)	(2,2)-(4,1)	(1,4)-(3,4)	(2,2)-(2,4)
(1,3)-(3,1)	(2,2)-(4,3)	(2,3)-(4,2)	(2,1)-(4,1)	(3,1)-(3,3)
(1,4)-(3,2)	(2,3)-(4,4)	(2,4)-(4,3)	(2,2)-(4,2)	(3,2)-(3,4)
(2,3)-(4,1)	(1,1)-(2,3)	(1,3)-(2,1)	(2,3)-(4,3)	(4,1)-(4,3)
(2,4)-(4,2)	(1,2)-(2,4)	(1,4)-(2,2)	(2,4)-(4,4)	(4,2)-(4,4)
	(2,1)-(3,3)	(2,3)-(3,1)		
	(2,2)-(3,4)	(2,4)-(3,2)		
	(3,1)-(4,3)	(3,3)-(4,1)		
	(3,2)-(4,4)	(3,4)-(4,2)		

q	r	
$d^2=2$	$d^2=1$	
(1,1)-(2,2)	(1,1)-(2,1)	(1,1)-(1,2)
(1,2)-(2,3)	(1,2)-(2,2)	(1,2)-(1,3)
(1,3)-(2,4)	(1,3)-(2,3)	(1,3)-(1,4)
(2,1)-(3,2)	(1,4)-(2,4)	(2,1)-(2,2)
(2,2)-(3,3)	(2,1)-(3,1)	(2,2)-(2,3)
(2,3)-(3,4)	(2,2)-(3,2)	(2,3)-(2,4)
(3,1)-(4,2)	(2,3)-(3,3)	(3,1)-(3,2)
(3,2)-(4,3)	(2,4)-(3,4)	(3,2)-(3,3)
(3,3)-(4,4)	(3,1)-(4,1)	(3,3)-(3,4)
	(3,2)-(4,2)	(4,1)-(4,2)
	(3,3)-(4,3)	(4,2)-(4,3)
	(3,4)-(4,4)	(4,3)-(4,4)

y	
$d^2=0$	
(1,1)-(1,1)	(3,1)-(3,1)
(1,2)-(1,2)	(3,2)-(3,2)
(1,3)-(1,3)	(3,3)-(3,3)
(1,4)-(1,4)	(3,4)-(3,4)
(2,1)-(2,1)	(4,1)-(4,1)
(2,2)-(2,2)	(4,2)-(4,2)
(2,3)-(2,3)	(4,3)-(4,3)
(2,4)-(2,4)	(4,4)-(4,4)

Table 2.2. Lists in each column all pairs of resolution cells situated in the same distance relationship. For example, all the resolution cell pairs listed under column c are related by distance $= \sqrt{10}$.

Resolution Cell	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)	(4,1)	(4,2)	(4,3)	(4,4)
(1,1)	y	r	k	d	r	q	j	c	k	j	i	b	d	c	b	a
(1,2)	r	y	r	k	q	r	q	j	j	k	j	i	c	d	c	b
(1,3)	k	r	y	r	j	q	r	q	i	j	k	j	b	c	d	c
(1,4)	d	k	r	y	c	j	q	r	b	i	j	k	a	b	c	d
(2,1)	r	q	j	c	y	r	k	d	r	q	j	c	k	j	i	b
(2,2)	q	r	q	j	r	y	r	k	q	r	q	j	j	k	j	i
(2,3)	j	q	r	q	k	r	y	r	j	q	r	q	i	j	k	j
(2,4)	c	j	q	r	d	k	r	y	c	j	q	r	b	i	j	k
(3,1)	k	j	i	b	r	q	j	c	y	r	k	d	r	q	j	c
(3,2)	j	k	j	i	q	r	q	j	r	y	r	k	q	r	q	j
(3,3)	i	j	k	j	j	q	r	q	k	r	y	r	j	q	r	q
(3,4)	b	i	j	k	c	j	q	r	d	k	r	y	c	j	q	r
(4,1)	d	c	b	a	k	j	i	b	r	q	j	c	y	r	k	d
(4,2)	c	d	c	b	j	k	j	i	q	r	q	j	r	y	r	k
(4,3)	b	c	d	c	i	j	k	j	j	q	r	q	k	r	y	r
(4,4)	a	b	c	d	b	i	j	k	c	j	q	r	d	k	r	y

Figure 2.9. Isotropic auto-covariance form.

Submatrix	Number of times submatrix occurs
S_1	$m_1 = K$
S_2	$m_2 = 2m_1-2$
S_3	$m_3 = m_2-2$
$\vdots$	
S_{K_r-1}	$m_{K_r-1} = m_{K_r-2}-2$
S_{K_r}	$m_{K_r} = m_{K_r-1}-2=2$

Figure 2.12 shows the form of a covariance matrix for an isotropic image partitioned into 3 row by 4 column subimages. With all this symmetry, we certainly should expect that the calculation of the eigenvectors of this matrix involves less work than we would need to do for a general covariance matrix. Perhaps if we are lucky, the transformation defined by the eigenvectors may even have a fast implementation.

5. Theory of Composite Matrices

The fast transform question is how close we can find a composite matrix to the general form of a autocovariance matrix. If we can find a composite matrix similar in form to the autocovariance matrix, then we would expect that the easily computed eigenvectors of the composite form would be similar to the eigenvectors of the autocovariance matrix and that the transformation of the image by these easily computed eigenvector (which have a direct product form) can be implemented as a fast transform.

Our situation is a fortunate one because the theory of composite matrices says that, if a matrix A can be partitioned into submatrices which have the same set of eigenvectors, then the eigenvectors of A can be formed as the direct product of the eigenvectors of the submatrices with the eigenvectors of the matrix of corresponding eigenvalues. Hence if:

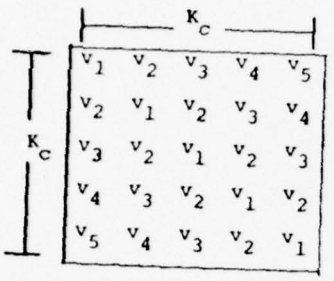


Figure 2.10. Illustrates a 5x5 submatrix of the covariance matrix of an isotropic image. The submatrix is toeplitz and has only five distinct entries: v_1, v_2, v_3, v_4, v_5.

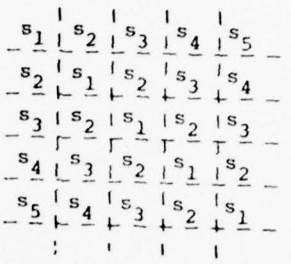


Figure 2.11. Illustrates how the covariance matrix of an isotropic image can be partitioned into submatrices each of the toeplitz form shown in Figure 2.9.

Figure 2.12. shows the form of a covariance matrix for an isotropic image partitioned into 3 row by 4 column subimages. With all this symmetry, we certainly should expect that the calculation of the eigenvectors of this matrix involves less work than we would need to do for a general covariance matrix. Perhaps if we are lucky, the transformation defined by the eigenvectors may even have a fast implement

i	g	f	g	h	e	f	e	d	c	b	a
g	i	g	h	g	h	e	f	e	b	c	b
f	g	i	e	h	g	d	e	f	a	b	c
g	h	e	i	g	f	g	h	e	f	e	d
h	g	h	g	i	g	h	g	h	e	f	e
e	h	g	f	g	i	e	h	g	d	e	f
f	e	d	g	h	e	i	g	f	g	h	e
e	f	e	h	g	h	g	i	g	h	g	h
d	e	f	e	h	g	f	g	i	e	h	g
c	b	a	f	e	d	g	h	e	i	g	f
b	c	b	e	f	e	h	g	h	g	i	g
a	b	c	d	e	f	e	h	g	f	g	i

$= \Sigma$

Figure 2.12. Isotropic auto-covariance form when an isotropic image is partitioned into 3 x 4 subimages $K_c = 3$ $K_r = 4$

$$A = \begin{pmatrix} A_1 & A_2 & A_3 \\ A_4 & A_5 & A_6 \\ A_7 & A_8 & A_9 \end{pmatrix}$$

and v is an eigenvector of each submatrix A_i with corresponding eigenvalue λ_i, and u is an eigenvector of the matrix

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_4 & \lambda_5 & \lambda_6 \\ \lambda_7 & \lambda_8 & \lambda_9 \end{pmatrix}$$

with corresponding eigenvalue n then: the direct product* uv is an eigenvector of A with corresponding eigenvalue n. [4] The covariance matrix for the isotropic image almost has the required property of the composite matrix. The covariance matrix can be partitioned into submatrices all of the same toeplitz form. However, this is not a guarantee that the submatrices have the same eigenvectors.

Empirical work with the submatrices of the covariance matrix of an isotropic image indicates that the submatrices are almost multiples of one another and therefore have almost the same eigenvector set. This justifies the following heuristic. Generate a submatrix with the property that the normed squares of the matrix of the differences between the best multiple of it and any given submatrix of the covariance matrix is the smallest when averaged over all submatrices. Then replace each submatrix with the best multiple of the generated submatrix. This creates a covariance matrix having the required composite structure. The eigenvectors can be determined

---

* If $u' = (u_1,\ldots,u_n)$ and $v' = (v_1,\ldots,v_m)$, then the direct product

$$(uv)' = (u_1v_1, u_1v_2, \ldots, u_1v_m, u_2v_2, \ldots, u_2v_m, \ldots, u_nv_1, u_nv_2, \ldots, u_nv_m)$$

quickly and the transformation defined by the eigenvectors has a fast implementation.

To demonstrate this approximation, consider the Isotropic Auto-covariance matrix in Figure 2.12. If we generate the second moment matrix considering each submatrix as a vector the following form emerges:

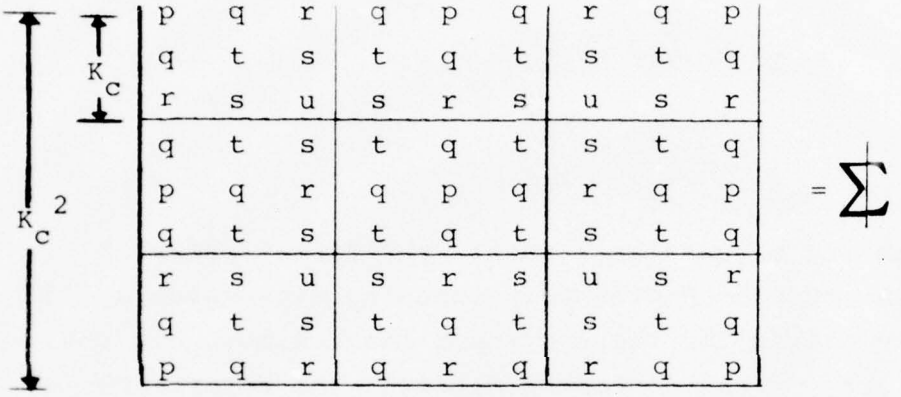


Figure 2.13. Form of the second moment matrix derived from the isotropic covariance matrix with each submatrix as a vector. All entries labeled q, for example, are the sums of products of entries labeled i and g, respectively, of the matrix in Figure 11.

Notice that all the distinct variables are in the first submatrix. It is therefore, possible to find the eigenvector having largest eigenvalue of this matrix without forming the entire matrix. All that is necessary is to store the first submatrix of the second moment matrix.

Let $v' = (v_1,\ldots,v_9)$ be the eigenvector of $\sum_1$ having largest eigenvalue λ. Then $\sum_1 v = \lambda v$. The first row of the matrix equation is:

$$pv_1 + qv_2 + rv_3 + qv_4 + pv_5 + qv_6 + rv_7 + qv_8 + pv_9 = v_1$$

and the fifth row is

$$pv_1 + qv_2 + rv_3 + qv_4 + pv_5 + qv_6 + rv_7 + qv_8 + pv_9 = v_5$$

Obviously, the first row and the fifth row are equal so $v_5 = v_1$. In like manner, the other rows may be compared, resulting in:

$$v_1=v_5=v_9 \qquad 3pv_1+4qv_2+2rv_3=\lambda v_1$$

$$v_2=v_4=v_6=v_8 \qquad 3qv_1+4tv_2+2sv_3=\lambda v_2$$

$$v_3=v_7 \qquad 2rv_1+4sv_2+2uv_3=\lambda v_3$$

Separating these simultaneous equations we have

$$\begin{bmatrix} 3p & 4q & 2r \\ 3q & 4t & 2s \\ 3r & 4s & 2u \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

This always happens because in the general case we have only K_c independent equations, all of which make up the first submatrix, of the second moment matrix, derived fron an isotropic covariance array in this manner. We need not worry about which elements are equal as long as our index follows the toeplitz form. By using the resulting eigenvector and symmetry the appropriate multiples for each submatrix of the auto-covariance matrix can be determined. The coefficients found by this procedure may be illustrated by:

$$c_1 = (v_1\ v2\ v_3\ v_2\ v_1\ v_2\ v_3\ v_2\ v_1)'(i\ g\ f\ g\ i\ g\ f\ g\ i)$$

or as $c_1 = (v_1 v_2 v_3)\begin{pmatrix} 3i \\ 4g \\ 2f \end{pmatrix}$ using just the first three elements.

Likewise:

$$c_2 = (v_1 v_2 v_3)\begin{pmatrix} 3g \\ 4h \\ 2e \end{pmatrix}$$

$$c_3 = (v_1 v_2 v_3)\begin{pmatrix} 3f \\ 4e \\ 2d \end{pmatrix}$$

$$c_4 = (v_1 v_2 v_3)\begin{pmatrix} 3c \\ 4b \\ 2a \end{pmatrix}$$

The composite matrix $\sum_c$ can then be immediately formed as:

$$\sum_c = \begin{bmatrix} c_1 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_2 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_3 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_4 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} \\ c_2 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_1 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_2 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_3 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} \\ c_3 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_2 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_1 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_2 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} \\ c_4 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_3 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_2 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} & c_1 \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{bmatrix} \end{bmatrix}$$

The composite matrix resulting from this operation has several properties which should be recognized: First, the sub-matrices of the composite matrix all have the same eigenvectors (they commute). Second, each submatrix is of the same toeplitz form. Third, corresponding to the shared eigenvectors V_i is the lambda matrix of corresponding eigenvalues,

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\ \lambda_5 & \lambda_6 & \lambda_7 & \lambda_8 \\ \lambda_9 & \lambda_{10} & \lambda_{11} & \lambda_{12} \\ \lambda_{13} & \lambda_{14} & \lambda_{15} & \lambda_{16} \end{pmatrix}$$

λ_i being the eigenvalue of $c_i \begin{pmatrix} v_1 & v_2 & v_3 \\ v_2 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{pmatrix}$

under the eigenvector V_i. Obviously, for each V_i the lambda matrices differ only by a multiplicative constant and so also share the same eigenvectors. The direct product of these two sets of eigenvectors are the eigenvectors of $\sum_c$. Since the eigenvectors of $\sum_c$ can be represented by the direct product of two sets of vectors, the eigenvector transformation has a fast implementation. (See Figures 2.14 and 2.15)

Once the fast implementation has been defined by the direct product relation of the shared eigenvectors of the submatrices and the shared eigenvectors of the lambda matrices we have created a set of basis vectors which may be used to define an orthogonal transformation on an image. (See Figures 2.16 and 2.17).

6. Computational Results

In terms relative to the transform coding of the image, what we have essentially done is to see under what conditions the Karhunen-Loeve transform is separable so that each sub-image can be transformed by first operating on its rows and then by columns. We determined that in order for this to happen the image had to be isotropic and we had to approximate the submatrices of the isotropic covariance matrix with submatrices having the same eigenvectors. In order to determine if the image isotropicity assumption and stationarity approximations were good ones for image data, we ran some experiments comparing the fast approximate K-L transform with the standard K-L transform. We found, as described in the following discussion, that the fast approximate K-L transform gives results better than the Hadamard, Fourier, and Slant. A 512 by 512, 6 bit digital image was compressed at ratios of 2 bits/pel, 1 bit/pel, and .5 bits/pel. A comparison was made between the Standard K-L transform, the Fast K-L transform, the Discrete Linear Basis, the Slant transform and the Discrete Cosine transform. The error criteria

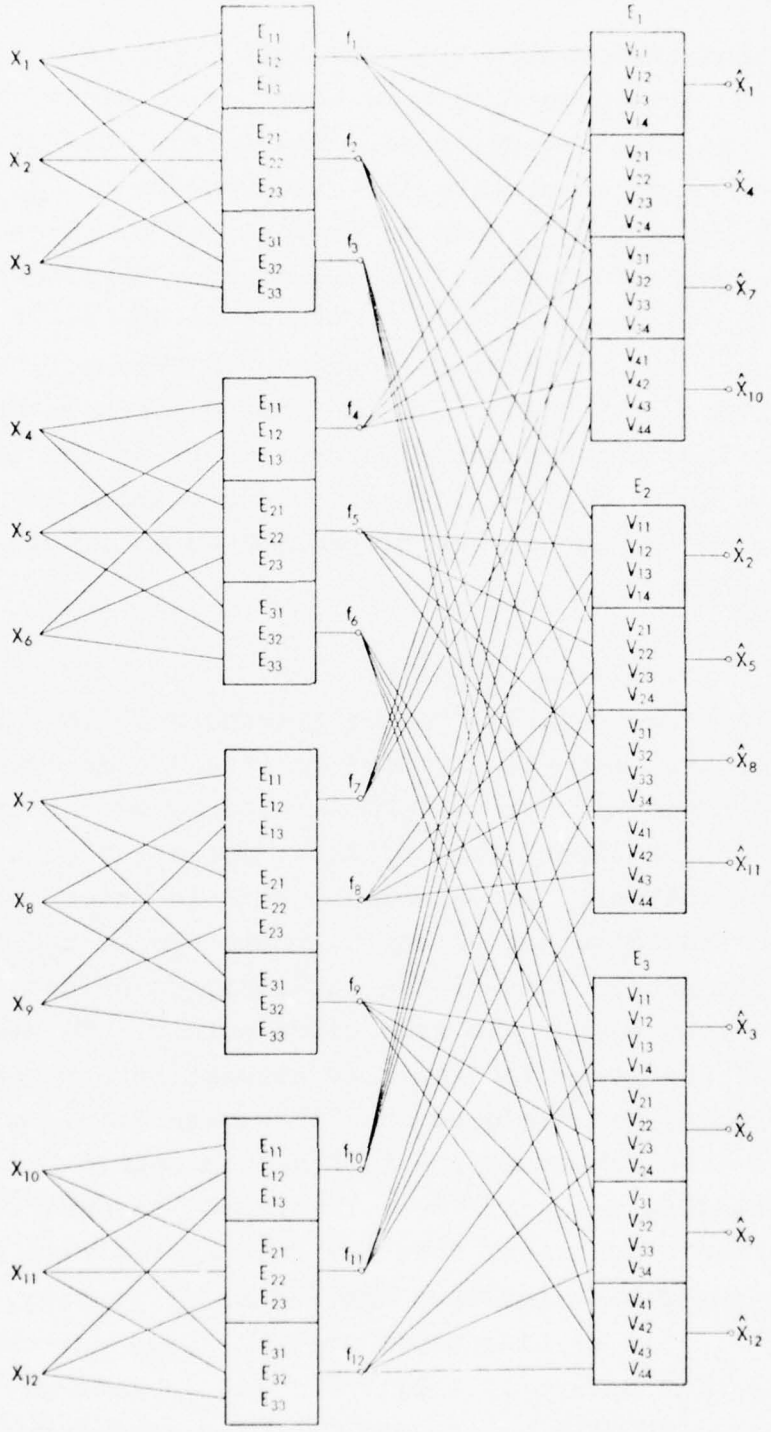


Figure 2.14. Each subimage has three rows by four columns. The fast implementation transforms each of the four rows of the subimage first and then transforms each of the three resulting columns.

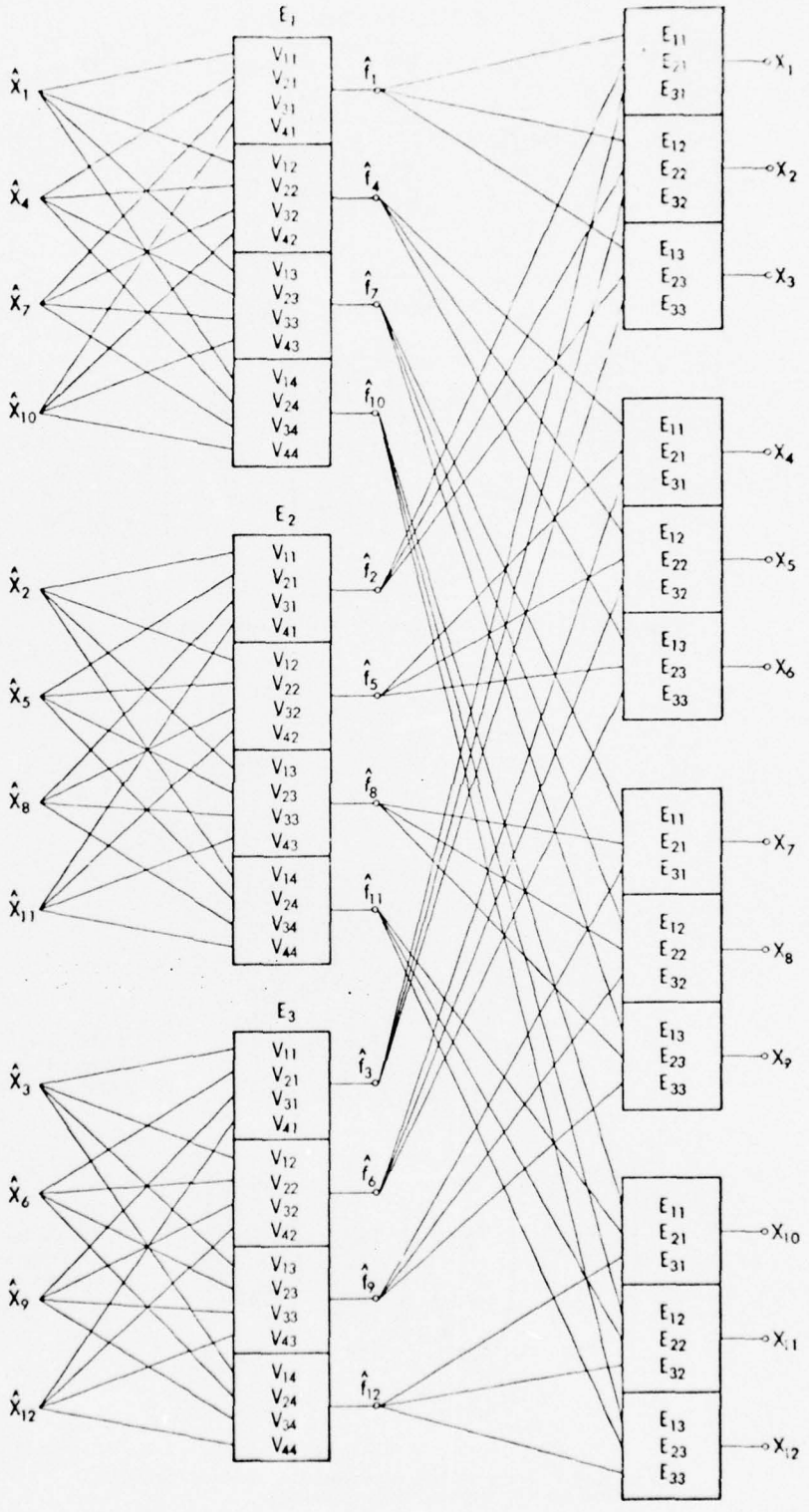


Figure 2.15. The inverse transform operates on each of the three columns and then operates on each of the four resulting rows.

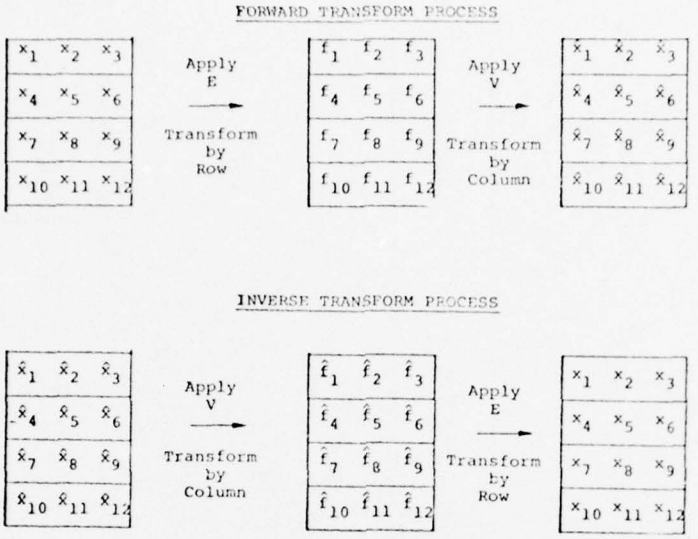


Figure 2.16. Row-column operation with basis vectors.

$$F = \left\{ \begin{pmatrix} E_{11} \\ E_{12} \\ E_{12} \end{pmatrix} \begin{pmatrix} E_{21} \\ E_{22} \\ E_{23} \end{pmatrix} \begin{pmatrix} E_{31} \\ E_{32} \\ E_{33} \end{pmatrix} \right\}$$

orthogonal raw transformation basis vectors

$$V = \left\{ \begin{pmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \end{pmatrix} \begin{pmatrix} v_{21} \\ v_{22} \\ v_{23} \\ v_{24} \end{pmatrix} \begin{pmatrix} v_{31} \\ v_{32} \\ v_{33} \\ v_{34} \end{pmatrix} \begin{pmatrix} v_{41} \\ v_{42} \\ v_{43} \\ v_{44} \end{pmatrix} \right\}$$

orthogonal column transformation basis vectors

Figure 2.17. Basis vector sets derived from composite matrix for direct product implementation.

chosen was the RMS and RMS correlated measures (Haralick and Shanmugam, 1974). [5]

The first error investigated was that of the stationarity and isotropic assumptions. The L_2 matrix norm of the differences between the various combinations are:

Auto-Covariance - Stationary Covariance	= 5.7628
Stationary Covariance - Isotropic Covariance	= 5.5923
Auto-Covariance - Isotropic Covariance	= 4.4510

The RMS errors should be compared to 401.765, the average value of the diagonal elements of the covariance matrices. If we let A, S, L represent the auto-covariance, of the original image, the stationary covariance, and isotropic covariance, respectively, then:

$$(A-L) = (A-S) - (L-S)$$

$$\|A-L\| \leq \|A-S\| + \|L-S\|$$

and our results satisfy the triangular inequality. In terms of distance measure Figure 2.18 depicts the differences of these approximations. The percentage difference, from the average variance of the auto-covariance, is 1.434% for the stationary assumption and 1.130% for the isotropic assumption. Examples of the actual auto-covariance matrix and auto-covariance matrices under the stationarity and isotropicity assumption are included in Figures 2.19, 2.20 and 2.21, respectively.

The eigenvectors of the standard auto-covariance matrix do not, in general, have the discrete sequency property which other fast transforms have. The eigenvectors resulting from the fast K-L do have the sequency property as the other fast transforms have. The eigenvectors for both the standard K-L and Fast K-L are compared in Figures 2.22 and 2.23. The subimage size used was 4 x 4 only because of minicomputer

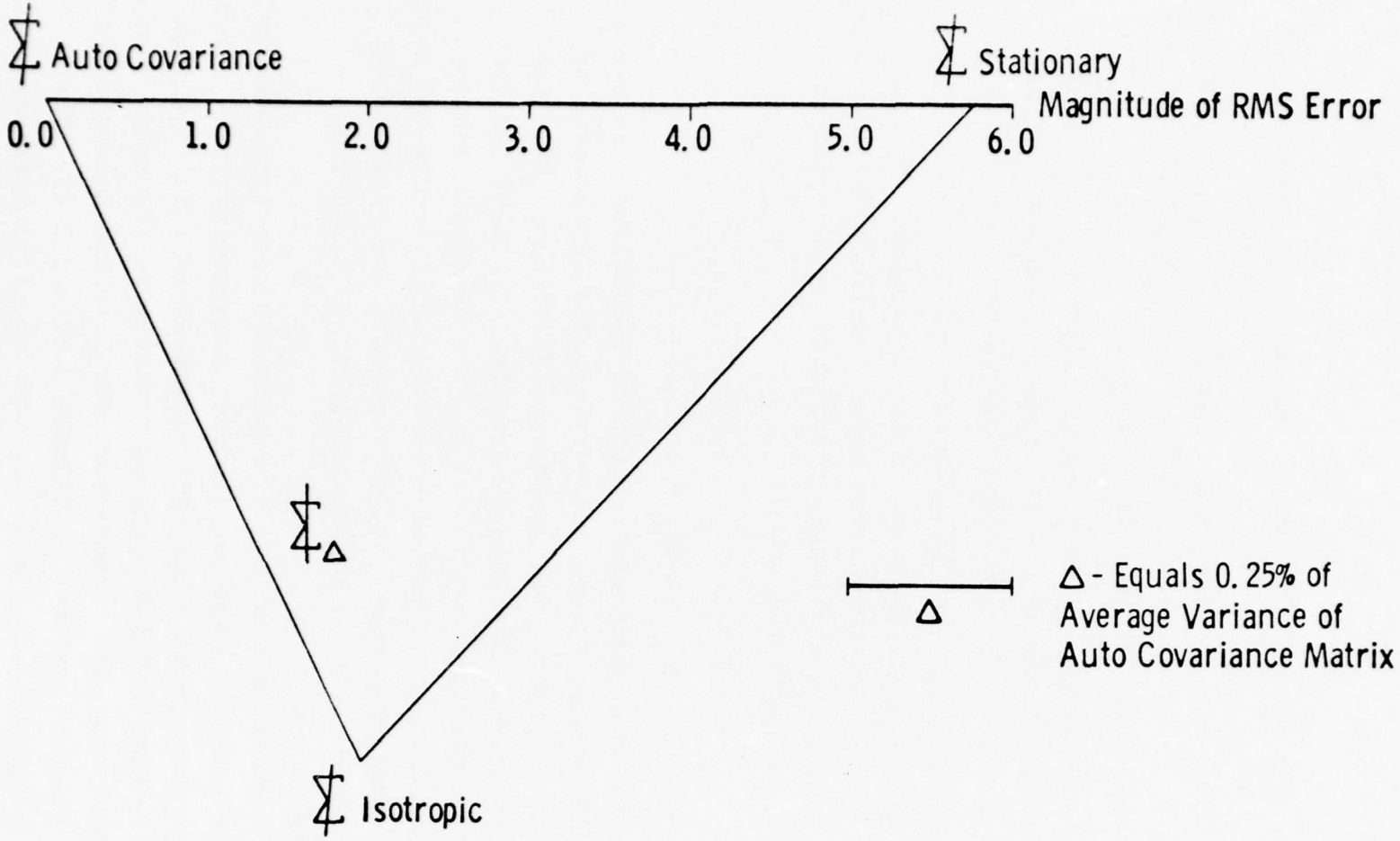


Figure 2.18. Illustrates the distance between the actual auto-covariance matrix derived from original image (upper triangle), the covariance matrix assuming image stationarity and the covariance matrix assuming image isotropicity.

401.757	383.327	370.692	361.734	379.608	372.763	364.253	356.036	365.537	361.459	354.369	347.887	356.392	352.230	346.560	341.021	
	401.384	383.142	370.850	373.542	379.508	372.179	363.446	361.543	365.361	360.092	353.700	353.119	356.311	352.258	345.568	
		399.991	382.111	364.907	373.654	378.506	371.421	355.281	361.578	363.734	359.484	347.753	353.029	355.857	350.502	
			398.535	357.480	364.666	373.308	377.281	349.812	355.731	360.737	362.931	343.394	348.069	352.996	353.300	
				401.457	384.079	372.026	362.976	380.793	374.374	365.442	358.569	367.994	363.172	356.967	350.775	
					401.232	383.675	371.883	375.629	380.965	373.366	365.432	364.721	367.873	363.196	356.167	
						400.745	383.357	366.884	375.017	379.598	373.212	358.967	364.989	367.346	362.653	
							399.955	359.253	366.115	374.280	379.014	352.356	358.484	364.376	365.338	
								403.033	387.230	374.571	366.432	383.538	376.529	368.225	360.772	
									404.067	385.970	374.307	378.707	381.648	376.457	367.586	
										402.801	385.748	368.920	377.935	383.012	375.718	
											402.767	381.521	369.193	379.023	381.941	
												403.088	385.209	373.348	364.143	
													403.017	385.181	373.437	
														402.687	385.654	
															400.782	

Figure 2.19. Auto-covariance matrix derived from original image (upper triangle).

401.801	384.426	372.673	363.811	380.581	373.757	365.495	357.971	366.498	362.080	355.877	349.747	356.304	352.293	346.786	341.426	
	401.801	384.426	372.673	380.556	380.581	373.757	365.495	366.471	366.498	362.080	355.877	356.273	356.304	352.293	346.786	
		401.801	384.426	380.543	380.556	380.581	373.757	366.455	366.471	366.498	362.080	356.251	356.273	356.304	352.293	
			401.801	380.518	380.543	380.556	380.581	366.425	366.455	366.471	366.498	356.227	356.251	356.273	356.304	
				401.801	384.426	372.673	363.811	380.581	373.757	365.495	357.971	366.498	362.080	355.877	349.747	
					401.801	384.426	372.673	380.556	380.581	373.757	365.495	366.471	366.498	362.080	355.877	
						401.801	384.426	380.543	380.556	380.581	373.757	366.455	366.471	366.498	362.080	
							401.801	380.518	380.543	380.556	380.581	366.425	366.455	366.471	366.498	
								401.801	384.426	372.673	363.811	380.581	373.757	365.495	357.971	
									401.801	384.426	372.673	380.556	380.581	373.757	365.495	
										401.801	384.426	380.543	380.556	380.581	373.757	
											401.801	380.518	380.543	380.556	380.581	
												401.801	384.426	372.673	363.811	
													401.801	384.426	372.673	
														401.801	384.426	
															401.801	

Figure 2.20. Auto-covariance matrix derived using stationary assumption.

401.801	382.504	369.586	360.058	382.504	377.156	368.647	361.764	369.586	368.647	361.166	354.802	360.058	361.764	354.802	348.827
	401.801	382.504	369.586	377.156	382.504	377.156	368.647	368.647	369.586	368.647	361.166	361.764	360.058	361.764	354.802
		401.801	382.504	368.647	377.156	382.504	377.156	361.166	368.647	369.586	368.647	354.802	361.764	360.058	361.764
			401.801	361.764	368.647	377.156	382.504	354.802	361.166	368.647	369.586	348.827	354.802	361.764	360.058
				401.801	382.504	369.586	360.058	382.504	377.156	368.647	361.764	369.586	368.647	361.166	354.802
					401.801	382.504	369.586	377.156	382.504	377.156	369.647	368.647	369.586	368.647	361.166
						401.801	382.504	368.647	377.156	382.504	377.156	361.166	368.647	369.586	368.647
							401.801	361.764	368.647	377.156	382.504	354.802	361.166	368.647	369.586
								401.801	382.504	369.586	360.058	382.504	377.156	368.647	361.764
									401.801	382.504	369.586	377.156	382.504	377.156	368.647
										401.801	382.504	368.647	377.156	382.504	377.156
											401.801	361.764	368.647	377.156	382.504
												401.801	382.504	369.586	360.058
													401.801	382.504	369.586
														401.801	382.504
															401.801

Figure 2.21. Auto-covariance matrix derived using isotropic assumption.

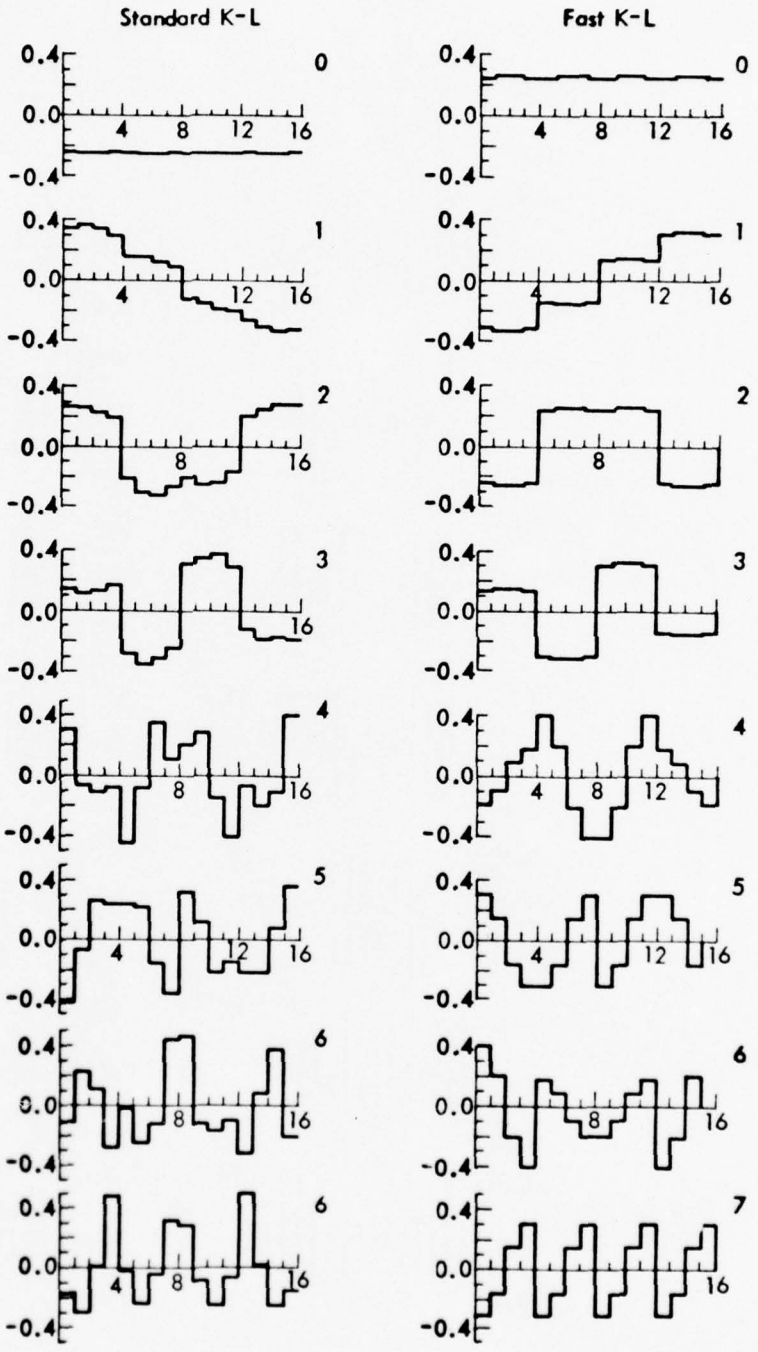


Figure 2.22. Comparison of eigenvectors for Standard K-L and Fast K-L in order of sequency.

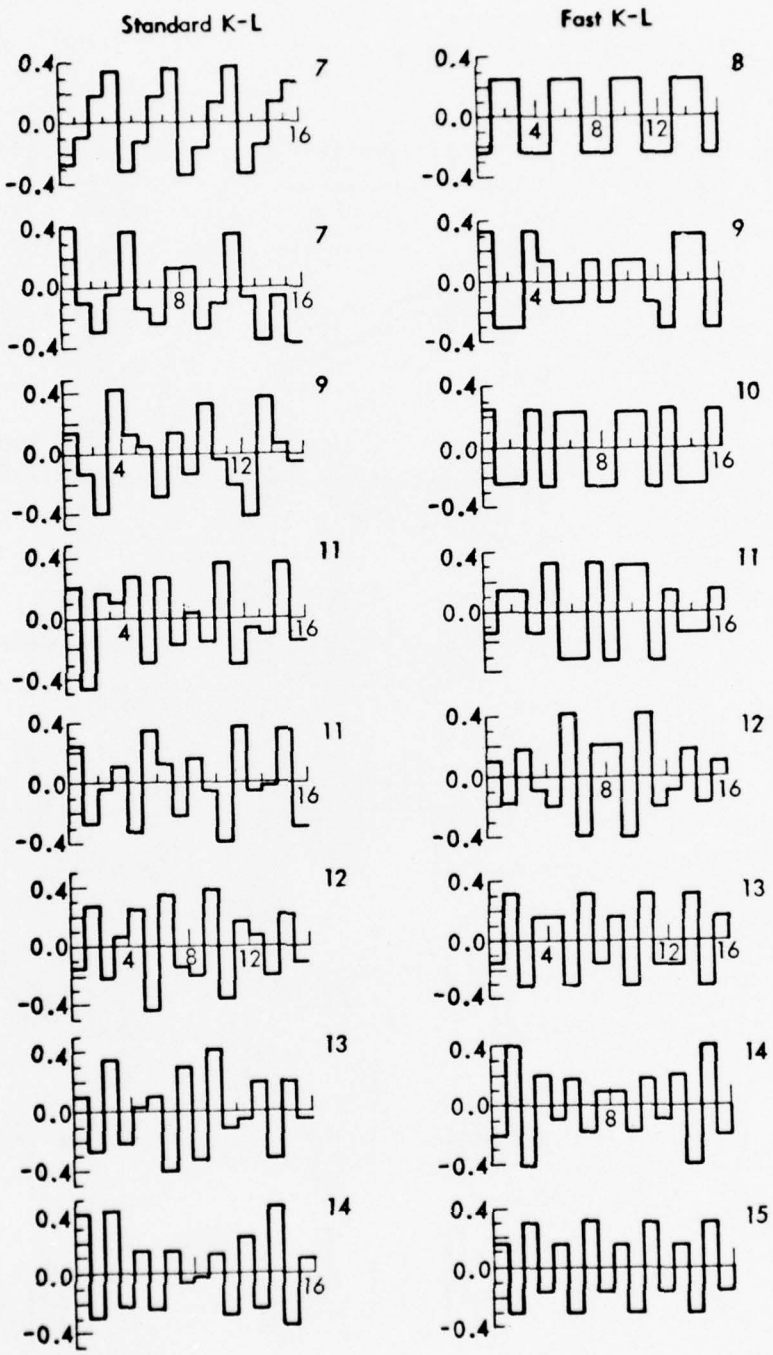


Figure 2.23. Comparison of eigenvectors for Standard K-L and Fast K-L in order of sequency.

memory and computational restrictions on determining the standard K-L. The fast version can use much larger subimages because of the symmetry and storage savings.

Results of the compression at 2 bits/pel, 1 bit/pel, and .5 bits/pel are plotted for comparison in Figure 2.24. It is apparent that the Fast K-L out-performs the other transforms and is closest to the optimum, differing from the optimum K-L only by approximately 1%. Table 2.3 gives a comparison of these errors.

7. Conclusions

In terms relative to the transform coding of an image we have presented the conditions under which the Karhunen-Loeve transform is separable and developed on approximate fast K-L transform. We have discussed the approximations of stationarity and isotropicity in terms of the L_2 distance norms. Our results indicate that the fast K-L transform is comparable to other discrete transforms both in terms of sequency and compression performance. Experimental data indicates that the fast K-L transform is closest to the optimum, differing from the optimum K-L by approximately 1%.

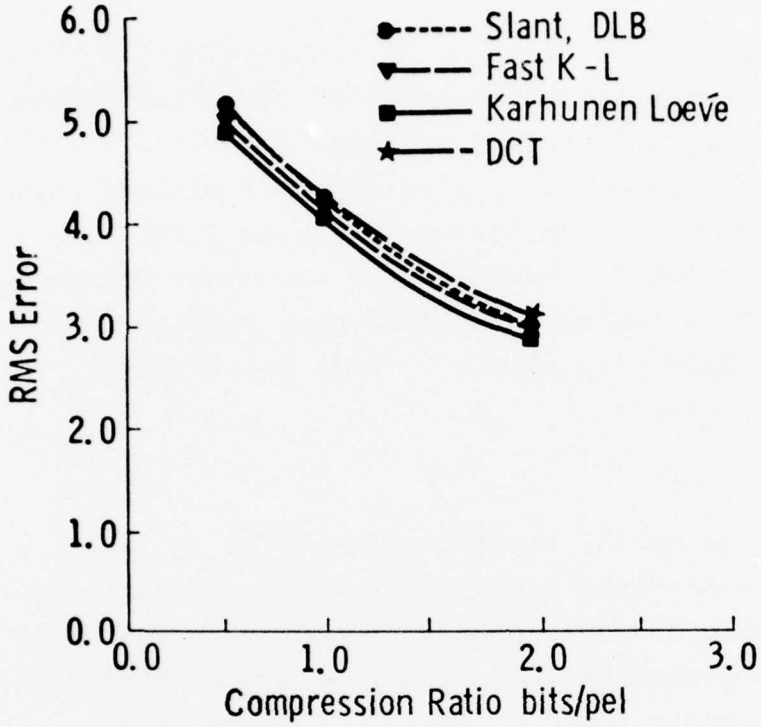


Figure 2.24a. Comparison of RMS error as a function of compression ratio for K-L, Fast K-L, Slant and DBL.

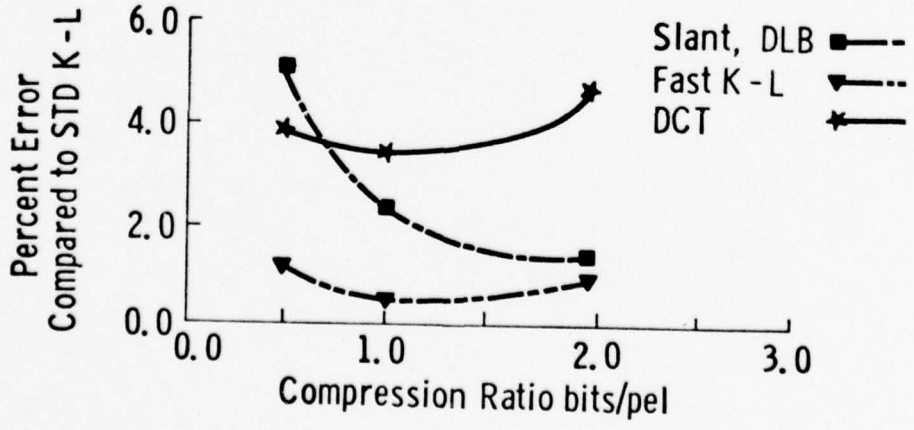


Figure 2.24b. Comparison of percentage error as a function of compression ratio for Fast K-L, Slant and DBL.

TABLE 2.3

TABLE OF RMS ERROR

TRANSFORM	COMPRESSION RATIO		
	2.0 bits/pel	1.0 bits/pel	.5 bits/pel
Standard K-L	2.9071	4.1855	4.9461
Fast K-L	2.9362	4.2098	5.0094
DLB	2.9492	4.2819	5.1993
DCT	3.0454	4.3288	5.0418
Slant	2.9492	4.2819	5.1993

% ERROR COMPARED TO STANDARD K-L

TRANSFORM	COMPRESSION RATIO		
	2.0 bits/pel	1.0 bits/pel	.5 bits/pel
Fast K-L	1.0000	.580	1.279
DLB	1.448	2.303	5.119
Slant	1.448	2.303	5.119
DCT	4.75%	3.423%	1.934%

Table 2.3 (Continued)

RMS CORRELATED ERROR BETWEEN STANDARD K-L AND FAST K-L

COMPRESSION RATIO

	DISTANCE	RMS ERROR	CORREL.	CORREL. ERROR	RMS ERROR	CORREL.	CORREL. ERROR	RMS ERROR	CORREL.	CORREL. ERROR
		2.0 bits/pel			1.0 bits/pel			.5 bits/pel		
STANDARD K-L	1	2.9071	.10238	.2976	4.1855	.12813	.5363	4.9461	.11094	.5487
	2	2.9071	.10202	.2966	4.1855	.12653	.5296	4.9461	.12055	.5963
	5	2.9071	.09966	.2897	4.1855	.12400	.5140	4.9461	.12901	.5980
	10	2.9071	.09400	.2733	4.1855	.12036	.5038	4.9461	.11870	.5871
FAST K-L	1	2.9362	.10415	.3058	4.2098	.10424	.4388	5.0094	.20142	1.0089
	2	2,9362	.10202	.2998	4.2098	.10509	.4424	5.0094	.19654	.9845
	5	2.9362	.10129	.2974	4.2098	.10862	.4572	5.0094	.19309	.9673
	10	2.9362	.10654	.2835	4.2098	.10613	.4468	5.0094	.18556	.9295

SECTION III

ENCODING FOR COMPRESSION

1. INTRODUCTION

In Chapter 1, the concept of the fast transforms were discussed and the criteria for compression of an image was given. Chapter 2 discussed under what conditions a fast optimum transform existed. In this chapter the quantizations and coding of the transformed coefficients is viewed from three categories. These categories are compression by sampling, compression by energy thresholding, and an optimum bit encoding scheme. This material is amplified by the suggestion of preprocessing. If information about the image, or class of images, statistics is known the image quality may be improved in terms of the visual criteria. Sampling techniques may be divided into two categories: those which are based on the unique structure of energy in the transform plane and those which are related to the spatial distribution of frequency/or sequency in the transform domain. Those techniques based on energy are guaranteed to give best results in terms of RMS error and they appear to have given the best results visually as well.

In this program the compression requirements are more severe than what has been reported in the literature. The compression in the transform domain must approach .6 and .7 bits/pel. (10:1 component compression). Of the methods reported in the literature the following are most predominant: [1]

a. Checkerboard sampling
b. Random sampling
c. Zonal sampling
d. Threshold sampling

To satisfy a given error criterion, the "best" compression maximizes the number of components set to zero while minimizing the error.

2. Statistical Measures

This section presents a brief discussion of several basic statistical measures derived from natural aerial imagery.

Probability Density Functions

In order to develop the marginal and joint probability density functions used consider the following:

Let $Z_x = \{1,\ldots N_x\}$ be the set of row indexes for the digital image I;

$Z_y = \{1,\ldots N_y\}$ be the set of column indexes for the digital image I.

$I: Z_x x Z_y \rightarrow S$

where S is the set of digitized grey levels

$s_1, s_2 \cdots s_K$.

Then the marginal probability is given by

$$P(s_i) = \frac{\#\{(n,m)\varepsilon(Z_x x Z_y) \mid I(n,m) = s_i\}}{\#Z_x x Z_y}$$

Let R be a binary relation defining the specified spatial relationship of any two resolution cells.

$R \subseteq (Z_x x Z_y) \times (Z_x x Z_y)$ defined by

$$R = \{((a,b),(c,d) \mid \rho((a,b),(c,d)) = r\}$$

where ρ is a metric on $Z_x x Z_y$.

The joint probability density function may then be defined by:

$$P(s_i,s_j) = \frac{\#\{((a,b),(c,d))\varepsilon R \mid I(a,b)=s_i,\ I(c,d)=s_j\}}{\#R}$$

Scene probability density functions are useful in selecting quantizer assignments in approaches which use additional quantizing assignments, linear or non-linear, prior to coding.

These histograms, of a two-dimensional image, indicate how much of the dynamic range of gray levels is used and give an estimate of the distribution. Ten images were received from the Air Force and had been digitzed to 6 bits (64 gray levels). A few of these images were selected at random and joint histograms made. This class of images were all military vehicles in natural settings and partial sky in the image. These estimates of the distributions are illustrated in Figures 3.1, 3.2, 3.3. In general, the relative joint distribution is a function of the distance between the grey tones. The high correlation in the images tested here show that a range of 2 and 5 with a circular pattern does not change the joint density distribution significantly (See Figures 3.4, 3.5). These statistics suggest that perhaps a better cosmetic quality could be achieved by requantizing the image prior to transforming, and compression. Indeed, this was verified as will be seen in Section 3.2 by the use of equal probability quantizing.

Equal probability quantization is a procedure by which the average information content of the quantized distribution is maximum. If we define $q_1, q_2, \ldots q_j$ as the quantized levels and the probabilities $p_1, p_2, \ldots, p_j$ associated with these quantization levels then the average information is defined to be:

$$H = - \sum_{i=1}^{j} P_i \log P_i$$

H will be maximum when the quantization levels are selected with equal probabilities. In order to determine these quantization levels the relative frequency function is determined for the gray levels of the pixels in the image. The cumulative frequency distribution is then formed from the relative frequency function and the probability range is divided into equal increments as a function of the gray levels.

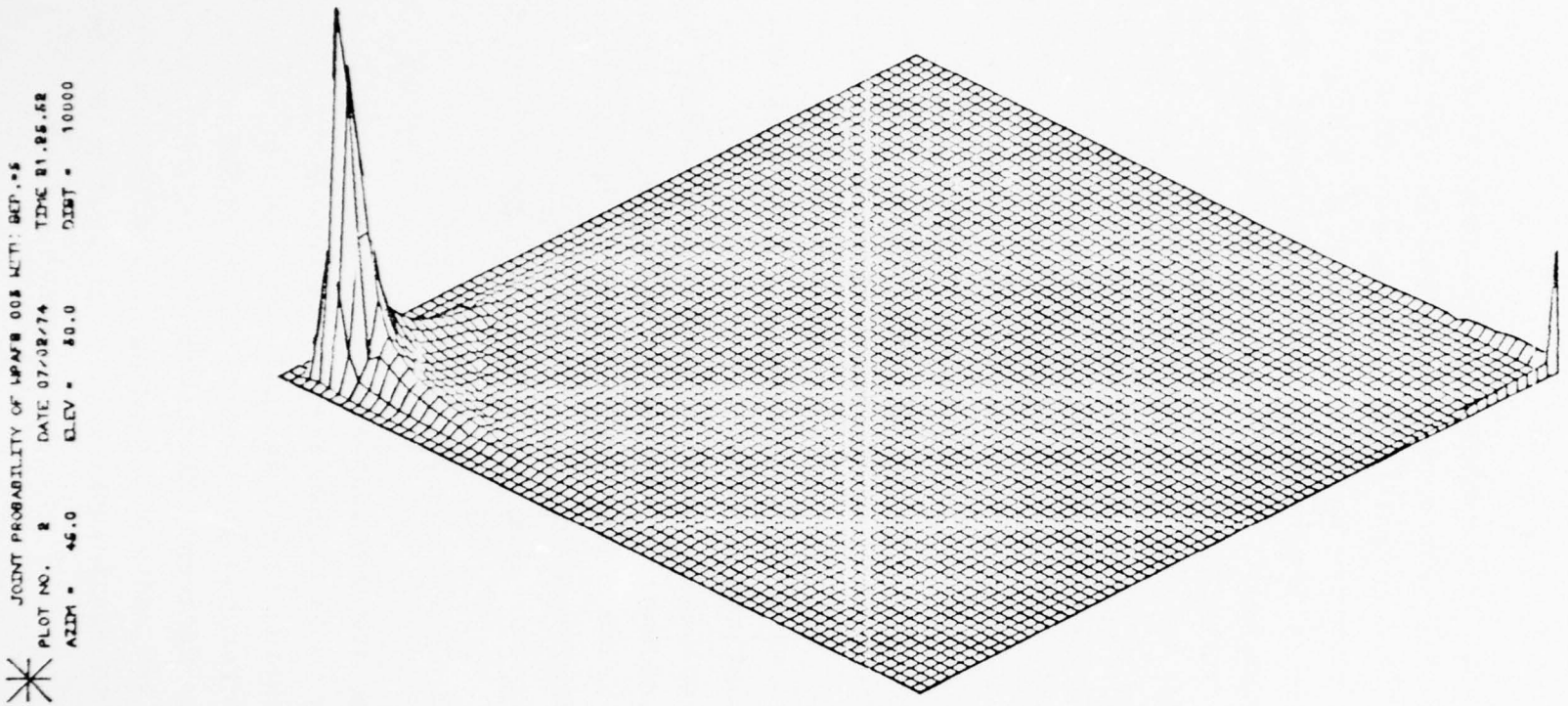


Figure 3.1. Joint Probability Distributions

JOINT PROBABILITY OF MPAFB 004 WITH DEP.=.6
PLOT NO. 8 DATE 07/04/74 TIME 12.38.08
AZIM = 45.0 ELEV = 30.0 DIST = 10000

Figure 3.2

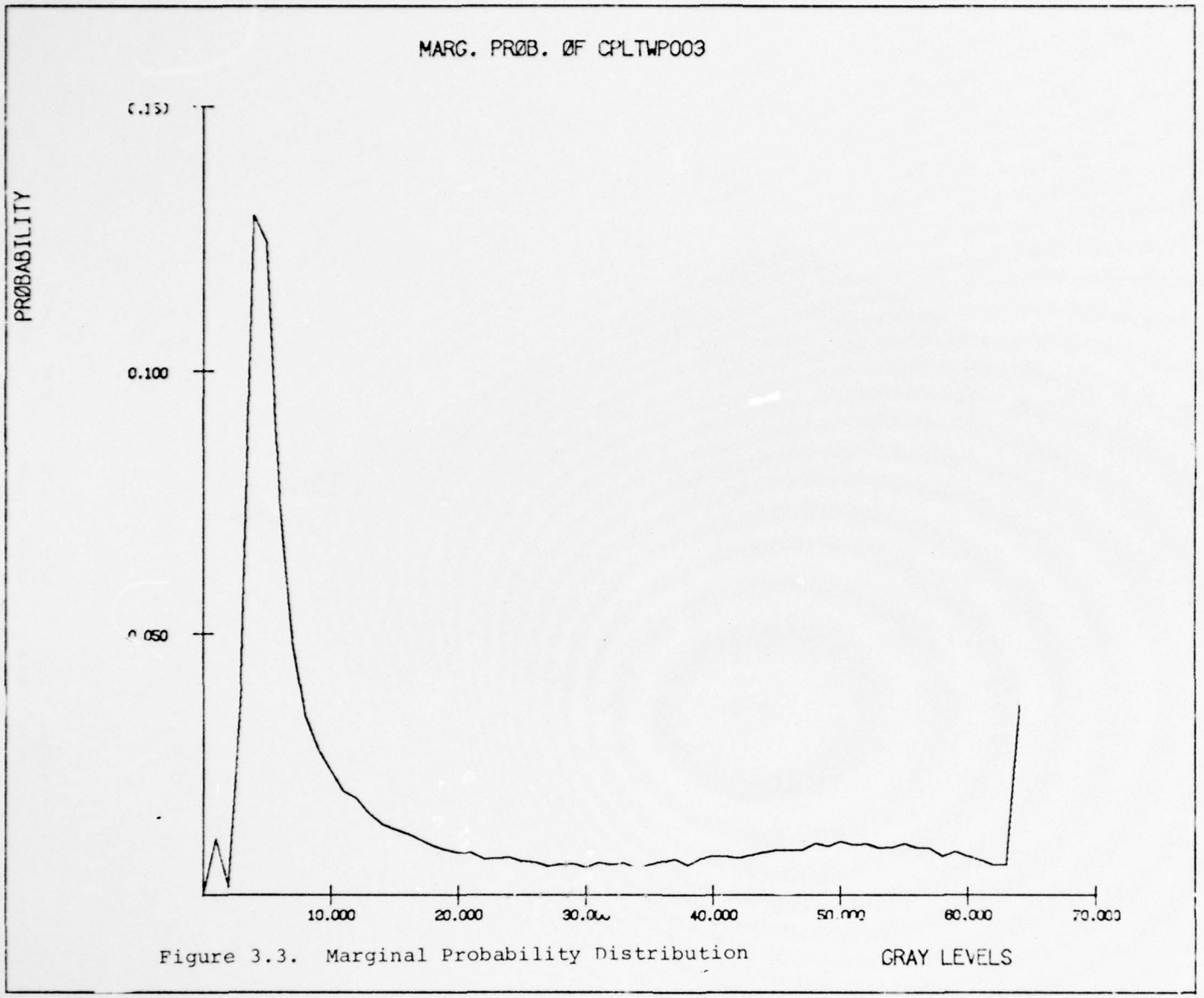


Figure 3.3. Marginal Probability Distribution

Figure 3.4. Joint Probability Distribution-spacing 5

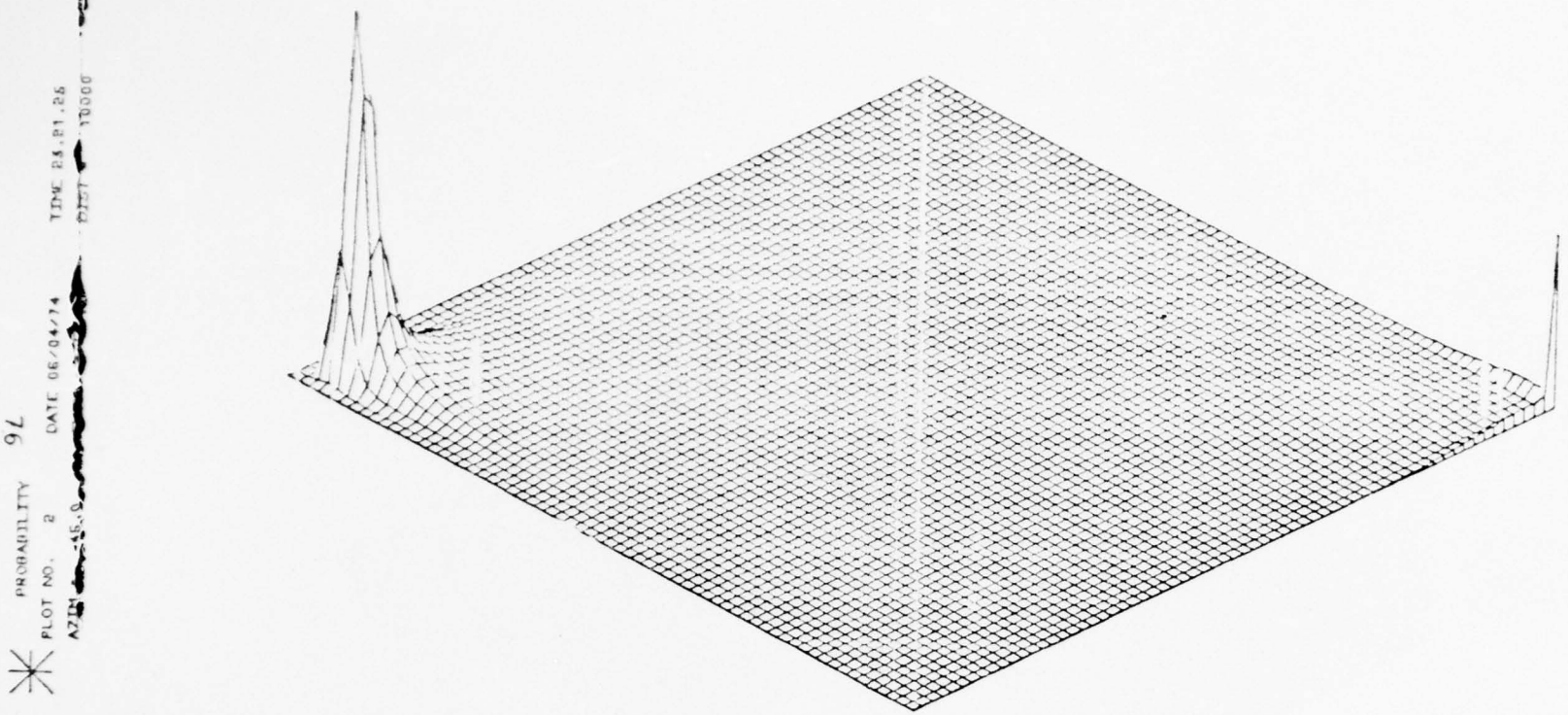


Figure 3.5. Joint Probability Distribution-spacing 2

3. Compression by Sampling

Sampling techniques may be divided into those which depend on the structure of energy and those depending on frequency of the component position. In some cases the two may be closely related but in general we cannot depend on this. However, if we are willing to make this assumption a great advantage is achieved by the ease of implementation. A notch filter (sampling) approach was, therefore, used to investigate the quality of the reconstructed image after compression.

Consider the transformed frequency space resulting from one of the transforms mentioned previously. The low frequencies of the transform domain contains the largest percent of energy. If an original image f(x,y) has magnitude ranging in unit steps from 0 to A then the maximum magnitude of a transform sample will be AN and the minimum non-zero magnitude $1/N^2$, where N is the number of samples. [1] Since the dynamic range is larger in the frequency domain only a few points can take on large values. This is because by Parseval's relation the energy in the spatial and transform domain are equal. A "notch" filter may then be constructed by transforming into the frequency domain and deleting frequencies in a specified range. The transmitted samples is the set of all remaining non-deleted frequencies. The frequencies to keep can be approximated by analysis of the smallest dimension which must be retained in the image. If this yields a recognizable target then frequencies can be eliminated and/or high frequencies emphasized. Figure 3.6 shows typical patterns used in the transform compression by "notch" filtering.

A set of three images were processed using this "notch filter" or sampling approach. The three originals had been digitized by the Avionics Laboratory, linearly to be 64

FREQUENCY COMPONENT F(u,v)

	F(0,0)	F(0,1)	F(0,2)	F(0,3)	F(0,4)	F(0,5)	F(0,6)	F(0,7)	F(-1,7)	F(-1,6)	F(-1,5)	F(-1,4)	F(-1,3)	F(-1,2)	F(-1,1)	F(-1,0)	
	INNER DISTANCE = 6 OUTER DISTANCE = 8																
F(0,0)	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	1.00	0.	0.	1.00	1.00	1.00	1.00	1.00	F(-1,0)
F(1,0)	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	1.00	0.	0.	1.00	1.00	1.00	1.00	1.00	F(-2,0)
F(2,0)	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	1.00	0.	0.	1.00	1.00	1.00	1.00	1.00	F(-3,0)
F(3,0)	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	1.00	0.	0.	0.	1.00	1.00	1.00	1.00	F(-4,0)
F(4,0)	1.00	1.00	1.00	1.00	0.	0.	0.	1.00	1.00	1.00	0.	0.	0.	1.00	1.00	1.00	F(-5,0)
F(5,0)	1.00	1.00	1.00	0.	0.	0.	0.	1.00	1.00	1.00	0.	0.	0.	0.	1.00	1.00	F(-6,0)
F(6,0)	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	F(-7,0)
F(7,0)	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	F(-8,0)
F(7,-1)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	F(-1,-8)
F(6,-1)	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	F(-1,-7)
F(5,-1)	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	F(-1,-6)
F(4,-1)	1.00	1.00	1.00	0.	0.	0.	0.	1.00	1.00	1.00	0.	0.	0.	0.	1.00	1.00	F(-1,-5)
F(3,-1)	1.00	1.00	1.00	1.00	0.	0.	0.	1.00	1.00	1.00	0.	0.	0.	1.00	1.00	1.00	F(-1,-4)
F(2,-1)	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	1.00	0.	0.	0.	1.00	1.00	1.00	1.00	F(-1,-3)
F(1,-1)	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	1.00	0.	0.	1.00	1.00	1.00	1.00	1.00	F(-1,-2)
F(0,-1)	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	1.00	0.	0.	1.00	1.00	1.00	1.00	1.00	F(-1,-1)

	INNER DISTANCE = 6 OUTER DISTANCE = 9																
	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	
	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	
	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	
	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	
	1.00	1.00	1.00	1.00	0.	0.	0.	0.	1.00	0.	0.	0.	0.	1.00	1.00	1.00	
	1.00	1.00	1.00	0.	0.	0.	0.	0.	1.00	0.	0.	0.	0.	0.	1.00	1.00	
	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	
	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	
	1.00	1.00	1.00	0.	0.	0.	0.	0.	1.00	0.	0.	0.	0.	1.00	1.00	1.00	
	1.00	1.00	1.00	1.00	0.	0.	0.	0.	1.00	0.	0.	0.	0.	1.00	1.00	1.00	
	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	
	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	
	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	

Figure 3.6. Notch Filtering Patterns

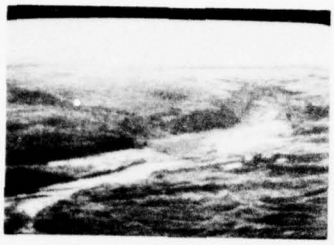
discrete levels. (See Figure 3.7) Figure 3.8 illustrates the sequence of operations.

The fast Fourier was then utilized in compression. The first stage of this process was to investigate which components would be deleted and what effect this deletion would have on the image. This was done without regard to blocking and without the log of brightness. The first notch filter used was that labeled as inner, outer range of (4,9). The components within the range of 4 and 9 being set to zero on each 16 by 16 block. This was done also for range (3,9), (4, 12), (3, 12). The corresponding compression ratio's are given below:

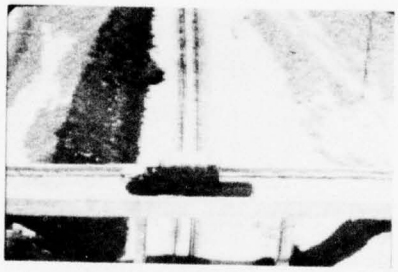
Notch Filter Range	Component Compression
4, 9	4:1
3, 9	5:1
4, 12	7:1
3, 12	12:1

Figure 3.9 shows the corresponding notch patterns.

The resulting images indicate that for higher component compression, the low frequency terms up to range 4 must be maintained. With a range of three, significant loss in resolution was noticed. The image comparison for these ranges are shown in Figure 3.10. The corresponding RMS error was also higher for low inner distance range of 3. This is because of a large change in energy. It was, therefore, decided to use notch filter with inner/outer range of (4, 9) and (4, 12). This produces better resolution. The (4, 12) notch deletes all frequencies beyond inner range 4. This did not degrade resolution when compared to (4, 9). The images were then compressed also using the log of the brightness and as expected and previously reported in the literature, the image at both 4:1 and 7:1 did appear to be better with respect to visual quality.



WPAFB 001
Tank



WPAFB 002
Truck

WPAFB 007
Aircraft

Figure 3.7. Original Images from Wright Patterson Air Force Base Digital Tape

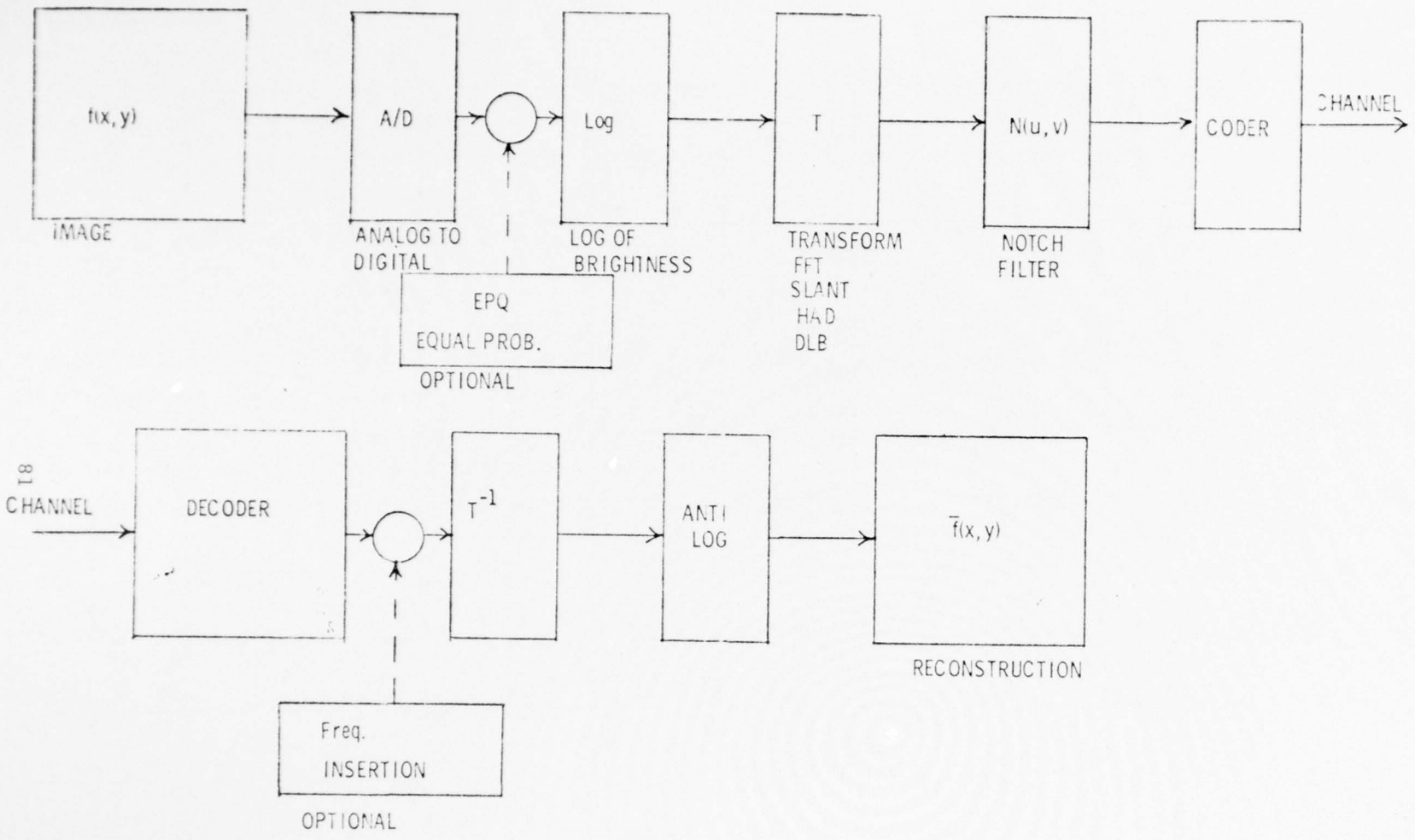


Figure 3.8. Sequence of Compression Procedure

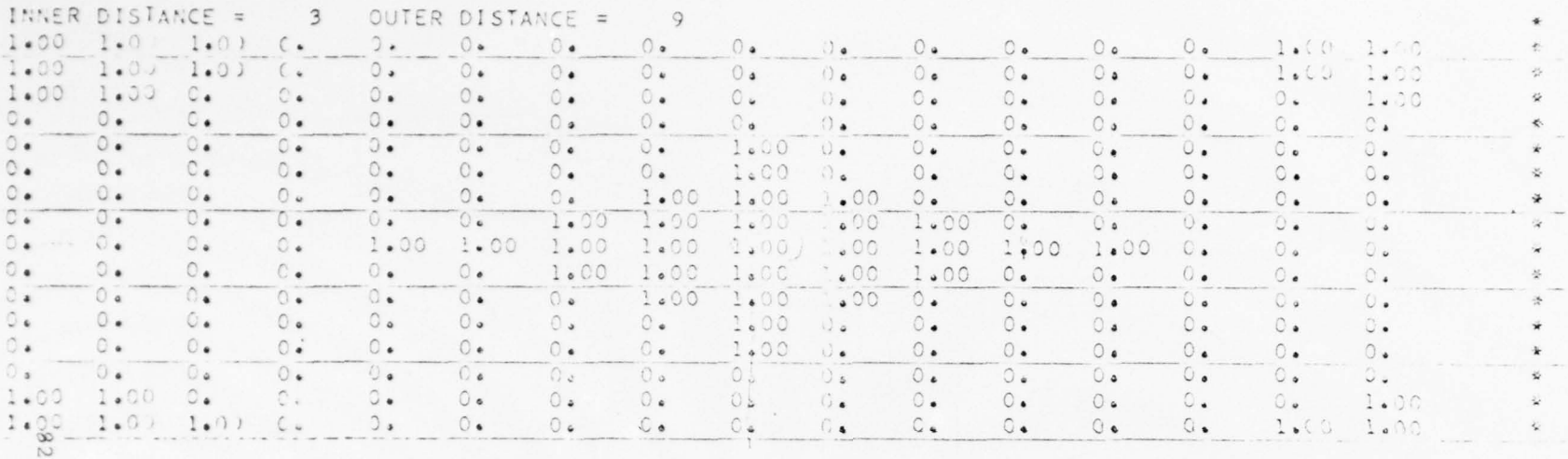
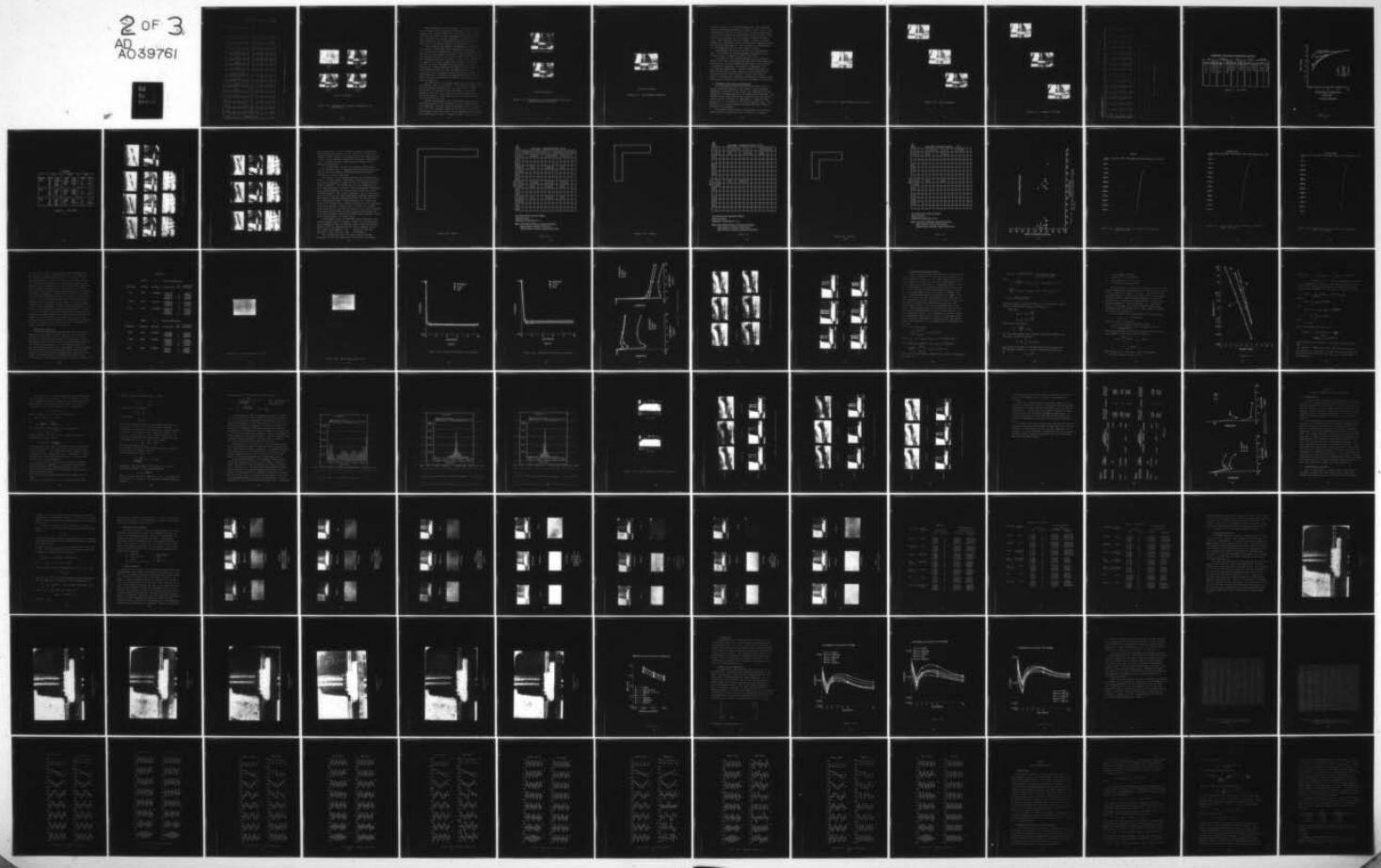


Figure 3.9a. Notch Patterns for 3,9 and 3,12

AD-A039 761 KANSAS UNIV/CENTER FOR RESEARCH INC LAWRENCE F/G 17/2
VIDEO BANDWIDTH COMPRESSION.(U)
FEB 77 N C GRISWOLD, R M HARALICK F33615-74-C-1093
UNCLASSIFIED 257-4 AFAL-TR-76-102 NL

2 OF 3
AD
A039761



BEST AVAILABLE COPY

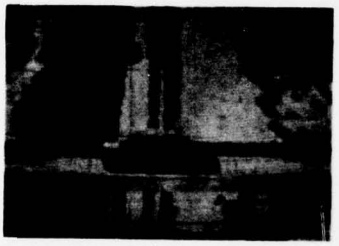
INNER DISTANCE = 4 OUTER DISTANCE = 9

1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	*
1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	*
1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	*
1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	*
0.	0.	0.	0.	0.	0.	0.	0.	1.00	0.	0.	0.	0.	0.	0.	0.	*
0.	0.	0.	0.	0.	0.	0.	0.	1.00	0.	0.	0.	0.	0.	0.	0.	*
0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	*
0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	*
0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	*
0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	*
0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	*
0.	0.	0.	0.	0.	0.	0.	0.	1.00	0.	0.	0.	0.	0.	0.	0.	*
0.	0.	0.	0.	0.	0.	0.	0.	1.00	0.	0.	0.	0.	0.	0.	0.	*
1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	*
1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	*
1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00	*

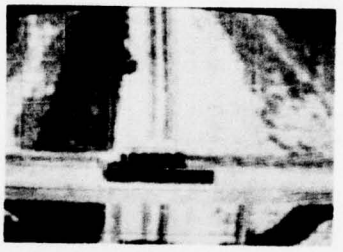
INNER DISTANCE = 4 OUTER DISTANCE = 12

1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00
1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00
1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00
1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00
1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00
1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.00	1.00	1.00

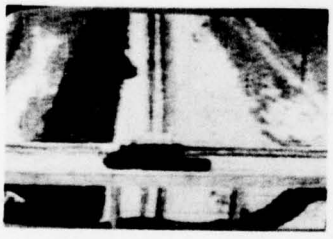
Figure 3.9b. Notch Patterns for 4,9 and 4,12



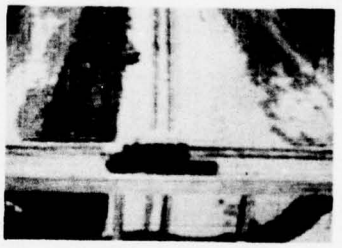
(3,9)
5:1



(3,12)
12:1



(4,9)
4:1



(4,12)
7:1

Fourier Transform

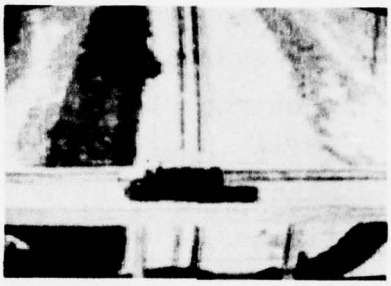
Figure 3.10. Comparison of Component Compression for Notch Filter

The blocking was then considered. The blocking occurs due to edge to edge brightness differences within the sub-images. This may be reduced by convolving a 2 x 2 window over the total image or by averaging the first and last row, column of the adjacent subimage. A comparison of the truck image with log of brightness and log of brightness with a 2 x 2 convolved window is shown in Figure 3.11.

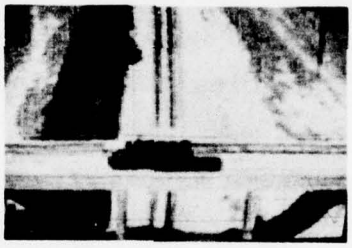
After studying the images, the question of improving the quality of the notch filter approach was examined. The (4, 9) notch (4:1 compression) filter was reprocessed but this time a high frequency weighing, linear proportioned to the magnitude of the d. c. term, was used. See Figure 3.12. This indicated that the image was, in fact, much improved in terms of the visual conception or cosmetic qualities. The RMS error was increased but this points out that by weighing the high frequencies the RMS error can be just as high because it does not make the image closer to the original in the mean square sense but still has better cosmetic qualities. Therefore, the decisions based on mean square error alone cannot be a complete description of how good the image is.

The RMS error of the bridge scene was calculated for the pre-emphasis case to be 6.53379. If the RMS error due solely to the blocking is evaluated it is found to be 4.72548. This error is a measure of the differences just on the boarders of our 16 x 16 blocks and is approximately 72% of the total error. This indicates with blocking removed the error would be 1.808.

Further improvement in the notch filter approach was believed possible if the statistics of the original image is considered. From the relative frequency plots of Figures 3.2, 3.3, it is apparent that most of the gray levels exist in the lower gray level range. Equal probability quantizing was, therefore, utilized to reduce the data from 6 bits to



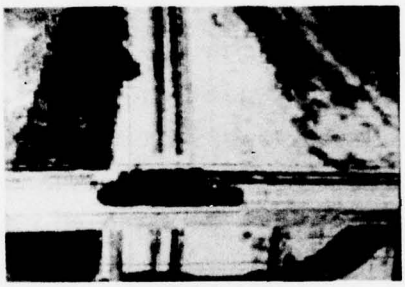
Log with Convolution



Log

Fourier Transform

Figure 3.11. Comparison of log of brightness and log of brightness with convolution



Fourier Transform

Figure 3.12. High Frequency Emphasis

5 bits to yield a 1.2 further compression. The resulting image was processed through the "notch" filter (4, 9). The compression of the (4, 9) notch is 4:1 with the additional 1.2 by equal probability quantization yielding a 4.8:1 compression, the results of this process is shown in Figure 3.13. This indicates that all images, if used with equal probability quantizing, can be increased in compression without noticable degradation. This results in a compression of 8-10 depending on the "notch" which is used.

The Hadamard and Slant transform were investigated next. The results of these transforms compressed by an L-notch filter is shown in Figures 3.14, 3.15.

An example of the L-notch pattern is illustrated in Figure 3.16. The truck scene was selected as the image with most activity and the Discrete Linear basis was also used in component comparison, and Table 3.1 and Figure 3.17 give the RMS comparisons. Since the DLB was the best in terms of RMS error it was compared to the Karhunen-Loeve the optimum under this criteria for the three scenes. Table 3.2 compares the component compression and error while the pictorial results are indicated in Figures 3.18 and 3.19.

4. Compression by Energy Thresholding

By investigating the energy associated with each frequency transform component, and sorting these frequency components according to that value, the importance of each component in reconstruction of the image may be deduced. The highest energy values being more important than the smaller values provide the means of compression. Almost all the energy will be contained in a few low frequency terms. The definition for energy component compression and process is found by the following:

The energy of projections corresponding to each basis vector used in a particular fast transform is computed.

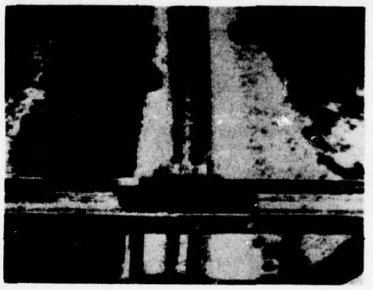
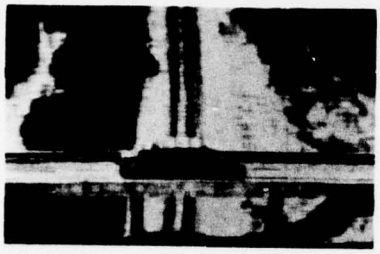
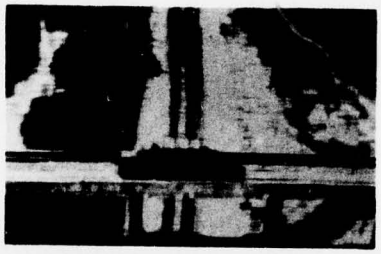


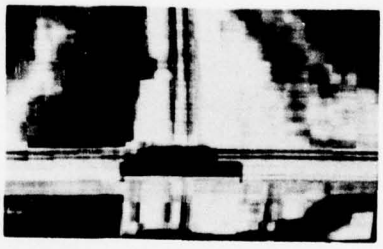
Figure 3.13. 32 Level - Equal Probability Quantizing



2:1

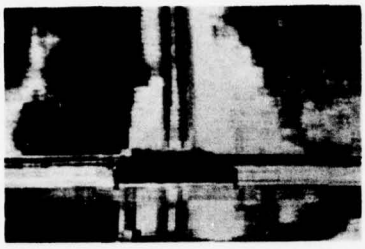


4:1

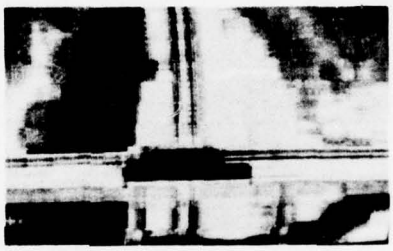


8:1

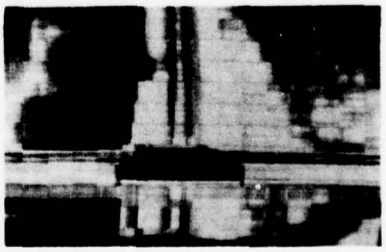
Figure 3.14. Slant Transform



2:1



4:1



8:1

Figure 3.15. Hadamard Transform

SEQUENCY COMPONENT S(u,v)

LENGTH = 10 WIDTH = 5

	S(0,0)	S(0,1)	S(0,2)	S(0,3)	S(0,4)	S(0,5)	S(0,6)	S(0,7)	S(0,8)	S(0,9)	S(0,10)	S(0,11)	S(0,12)	S(0,13)	S(0,14)	S(0,15)
0)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 3.16. L-Notch Filter Pattern

COMPRESSION - RMS ERROR SCENE 002 TRUCK ON BRIDGE

KL		FOURIER		DLB		SLANT		HADAMARD	
COMP	RMS ERROR	COMP	RMS ERROR	COMP	RMS ERROR	COMP	RMS	COMP	RMS
2:1	3.2556	3.87	5.712	2.13	4.008	2:1	4.381	2:1	5.66
4:1	4.696	5.12	6.649	5.33	5.374	4.26	4.862	4.26	6.88
8:1	6.129	7.1	5.88	8.0	5.953	8.26	6.358	8.26	6.90
16:1	7.176	12.11	6.8						

TABLE 3.1 - RMS ERRORS

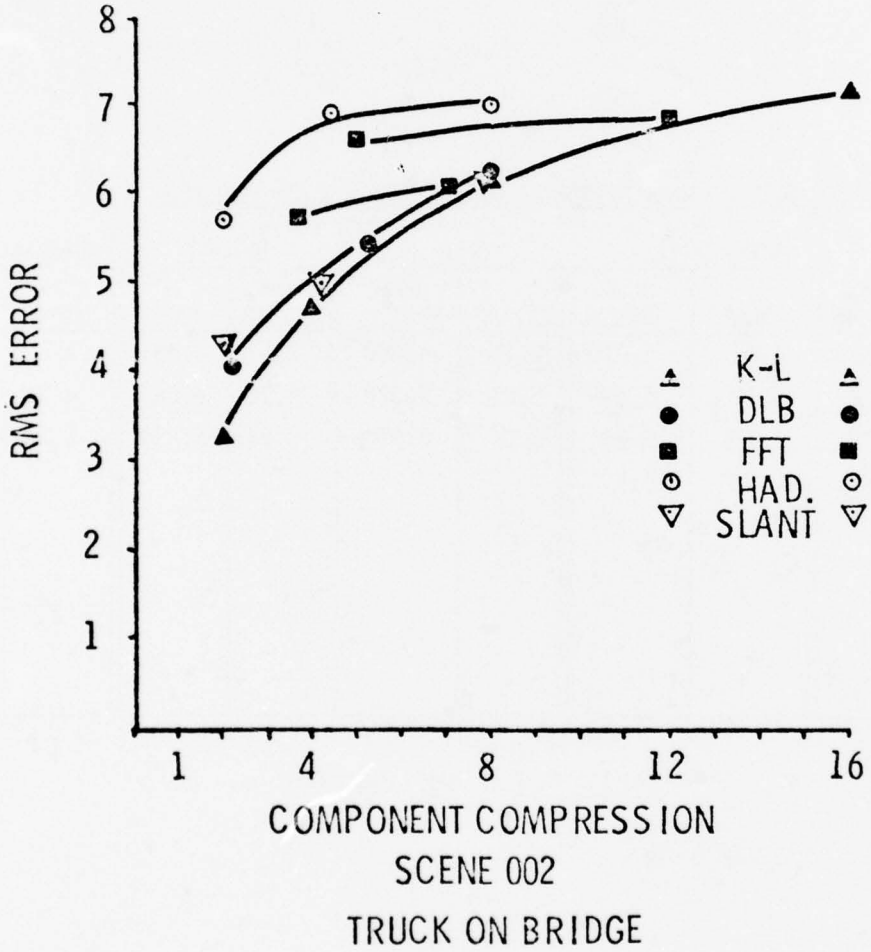

8
7
6
5
4
3
2
1
RMS ERROR
K-L
DLB
FFT
HAD.
SLANT
1
4
8
12
16
COMPONENT COMPRESSION
SCENE 002
TRUCK ON BRIDGE

Figure 3.17

3 - SCENES

	K-L		DLB		FFT	
SCENE#	COMP	RMS	COMP	RMS	COMP	RMS
001	2:1	2.2086	2.13:1	2.806	---	---
	4:1	3.22	3.55:1	3.139	---	---
	8:1	4.03	8.0:1	4.007	7.1:1	4.22
	16:1	5.056	---	---	---	---
002	2:1	3.255	2.13:1	4.008	---	---
	4:1	4.69	3.55:1	5.374	---	---
	8:1	6.12	8.0:1	5.953	7.1:1	6.13
	16:1	7.17	---	---	---	---
007	2:1	2.38	2.13:1	3.01	---	---
	4:1	3.77	3.55:1	4.35	---	---
	8:1	4.86	8:1	5.136	7.1:1	5.323
	16:1	6.81	---	---	---	---

TABLE 3.2 - RMS ERRORS

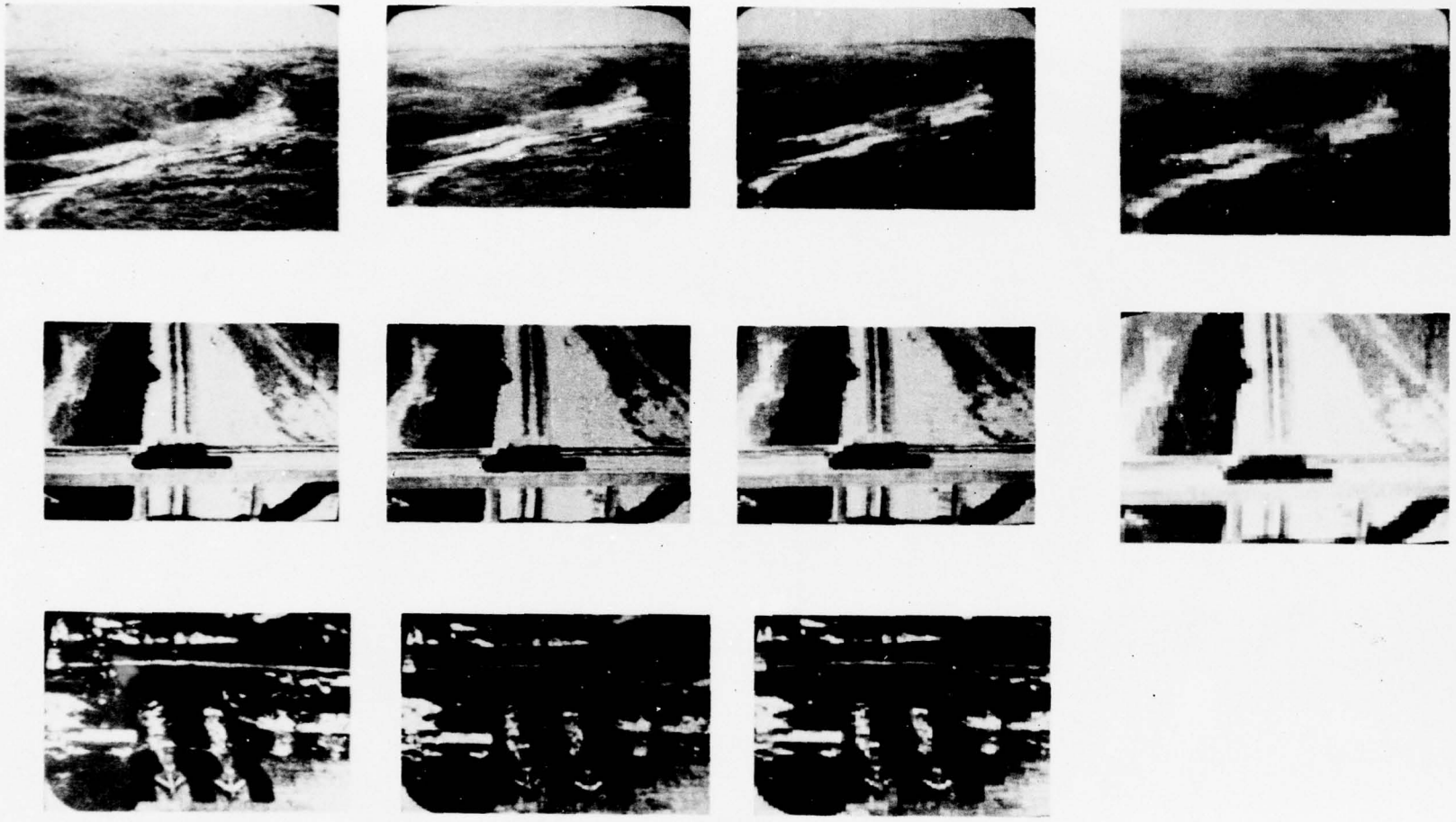


Figure 3.18. Compression for Three Scenes Karhunen-Loeve Transform

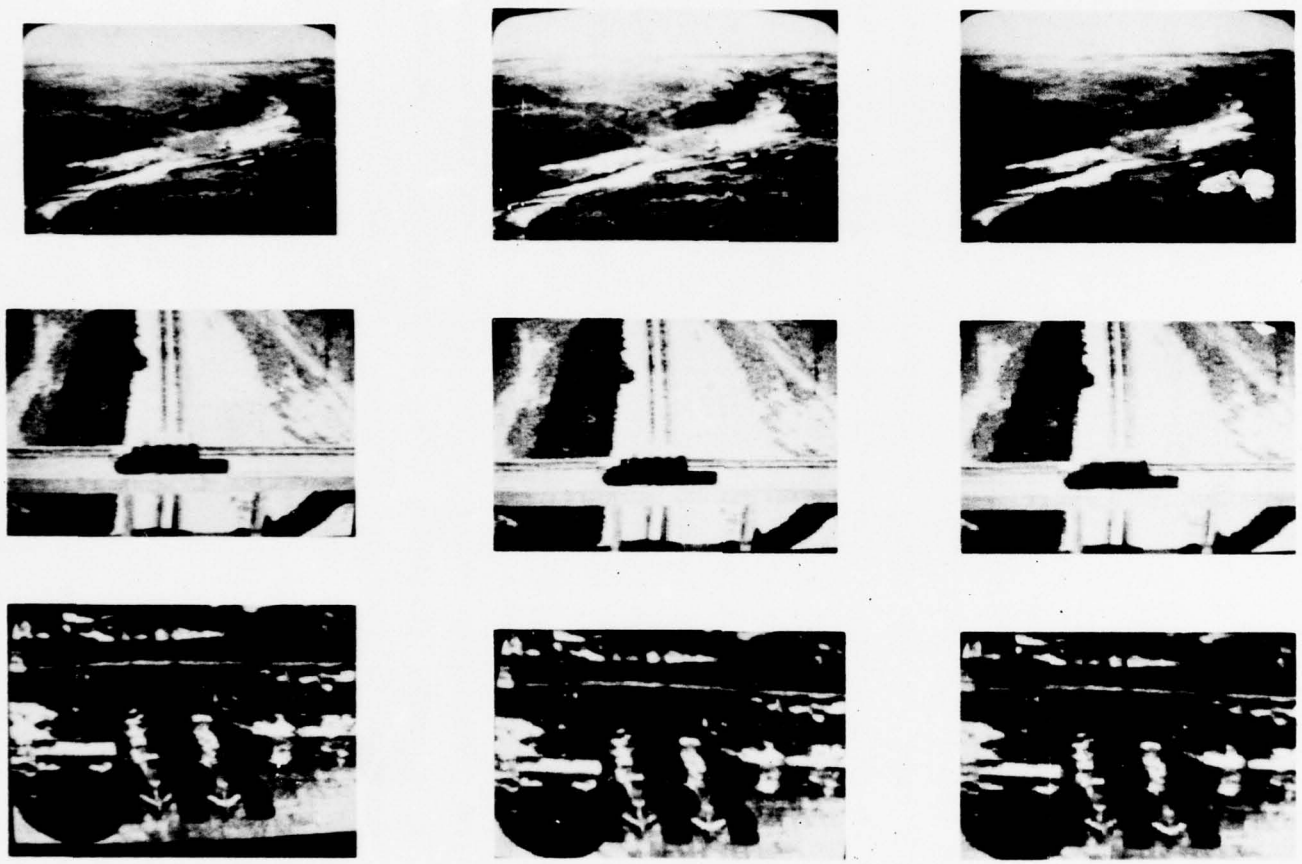


Figure 3.19. Discrete Linear Basis Transform

Those projections having less than a specified energy are modified by setting them to zero. The ratio of the total number of basis vectors or projections to the number not set to zero is the component compression achieved. Reconstruction is achieved by taking the inverse transform of the modified projections.

It is therefore, possible to improve on the earlier "notch" pattersn used in component compression by seeing how the positions of frequency components used are located according to energy importance.

The energy in the sampling approach was approximately 95-96%. In this energy thresholding technique energy values have been increased to a range of 99.3% to 99.94% of total energy available. The difference is in the components of the frequency domain selected. These projections (components) were ordered according to energy values calculated in the frequency domain. The first N components out of M possible were selected to yield the appropriate component compression. The energy in the M-N components account for the error in the reconstructed image. The patterns of an L-notch type filter used previously can then be compared to the components selected by energy. Figures 3.20, 3.21 and 3.22 compare these patterns with increasing energy for the slant transform. It becomes obvious that the notch patterns previously used are indiscriminately eliminating high energy component, therefore, contributing to a higher error.

From another view, the unsorted energy (variance) and sorted variance may be compared. Figure 3.23 illustrates a few components in the low and mid-range which are unsorted. Figures 3.24-3.26 demonstrate how this is changed for the DLB, SLANT, and Hadamard. Energy compression was then implemented using the DLB, SLANT, and Hadamard.

The corresponding RMS (root mean square) and RMS correlated errors are tabulated for the three transforms.

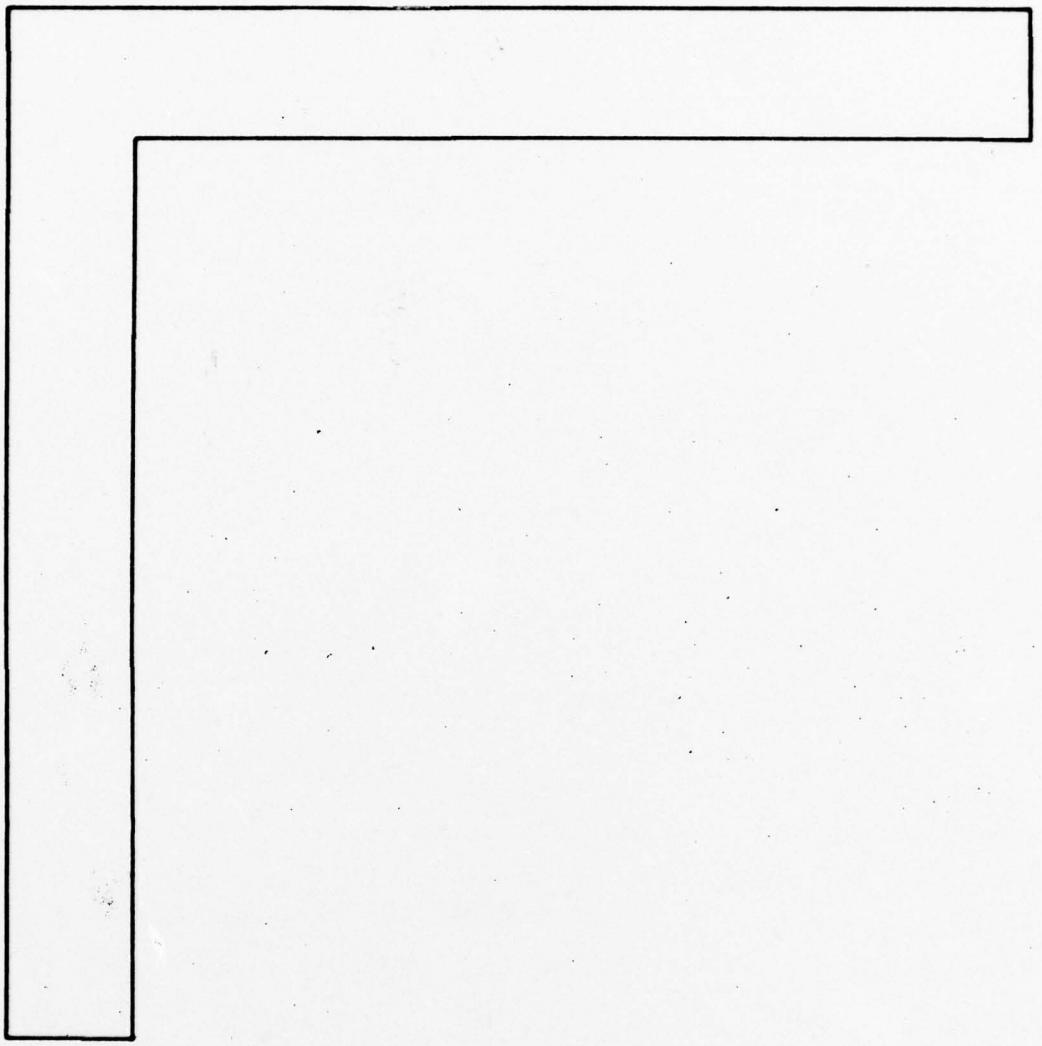


Figure 3.20. Overlay

DC Term

Sub-image Frequency Position F(1, V)

F(U, 1)															
1	2	34	57	8	19	25	52	4	7	36	58	13	18	29	64
3	6			20	44			11	21	62		28	41	63	
27															
61															
10	30			40				38	45			51			
17	54														
32															
49															
5	15			26	59			16	31			37	48		
9	24			43				33	42			60			
23															
56															
12	35			47				39	55			53			
14	50														
22															
46															

Two-dimensional Frequency Pattern
Slant Transform
Component Compression 4 : 1

Note: Each Block Indicates a Frequency Position.
Each Number Indicates Energy Order.
Blank Position Indicates Component Not Used.

Figure 3.20.

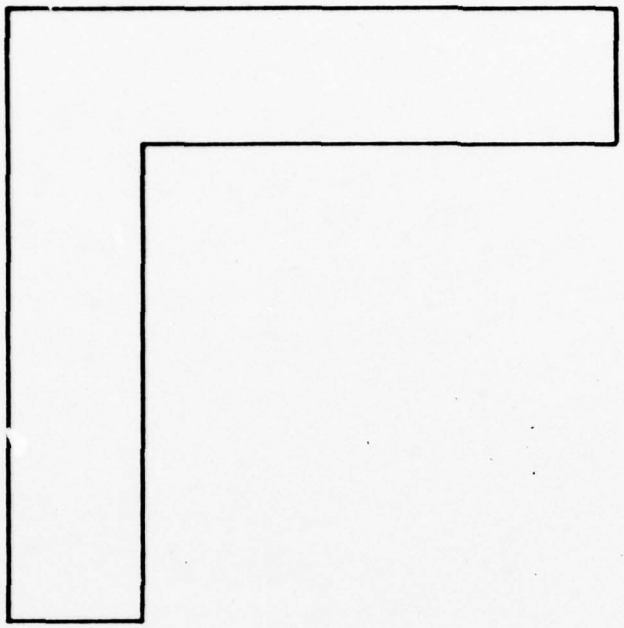


Figure 3.21. Overlay

DC Term

Sub-image Frequency Position F(1, V)

F(U, 1)

1	2			8	19	25		4	7			13	18	29	
3	6			20				11	21			28			
27															
10															
17															
31															
5	15			26			30	16							
9	24														
23															
12															
14															
22															

Two-dimensional Frequency Pattern
Slant Transform
Component Compression 8 : 1

Note: Each Block Indicates a Frequency Position.
Each Number Indicates Energy Order.
Blank Position Indicates Component Not Used.

Figure 3.21

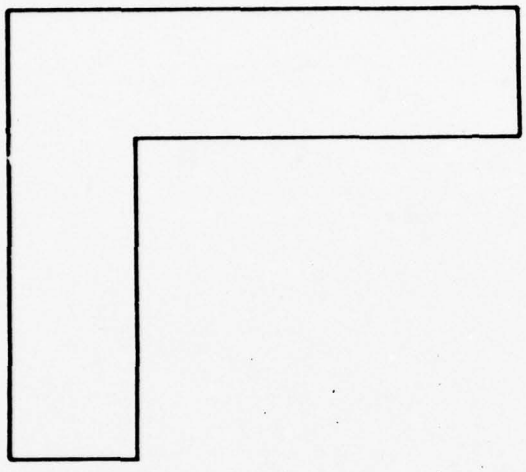
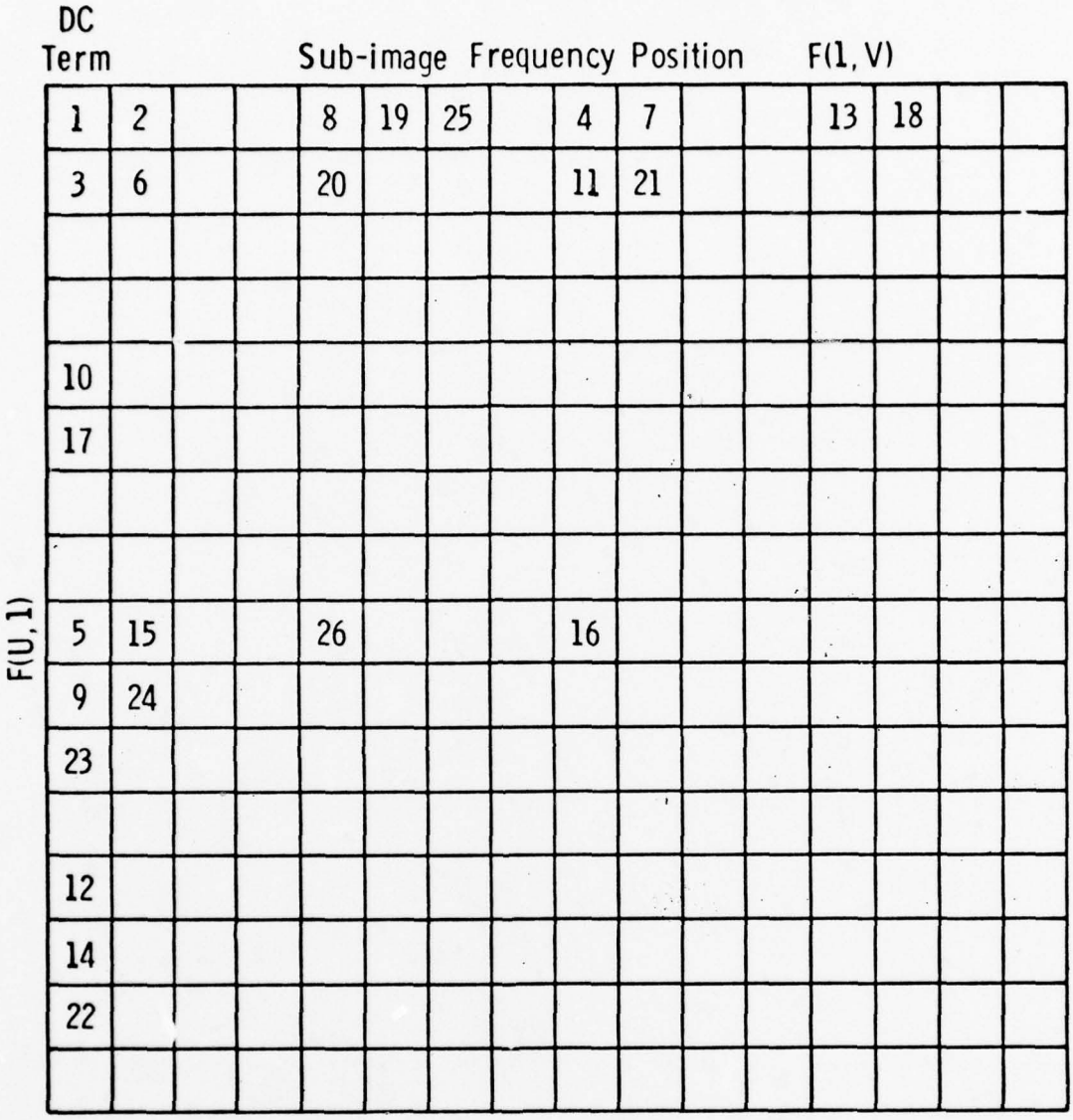


Figure 3.22. Overlay



Two-dimensional Frequency Pattern
Slant Transform
Component Compression 10 : 1

Note: Each Block Indicates a Frequency Position.
Each Number Indicates Energy Order.
Blank Position Indicates Component Not Used.

Figure 3.22

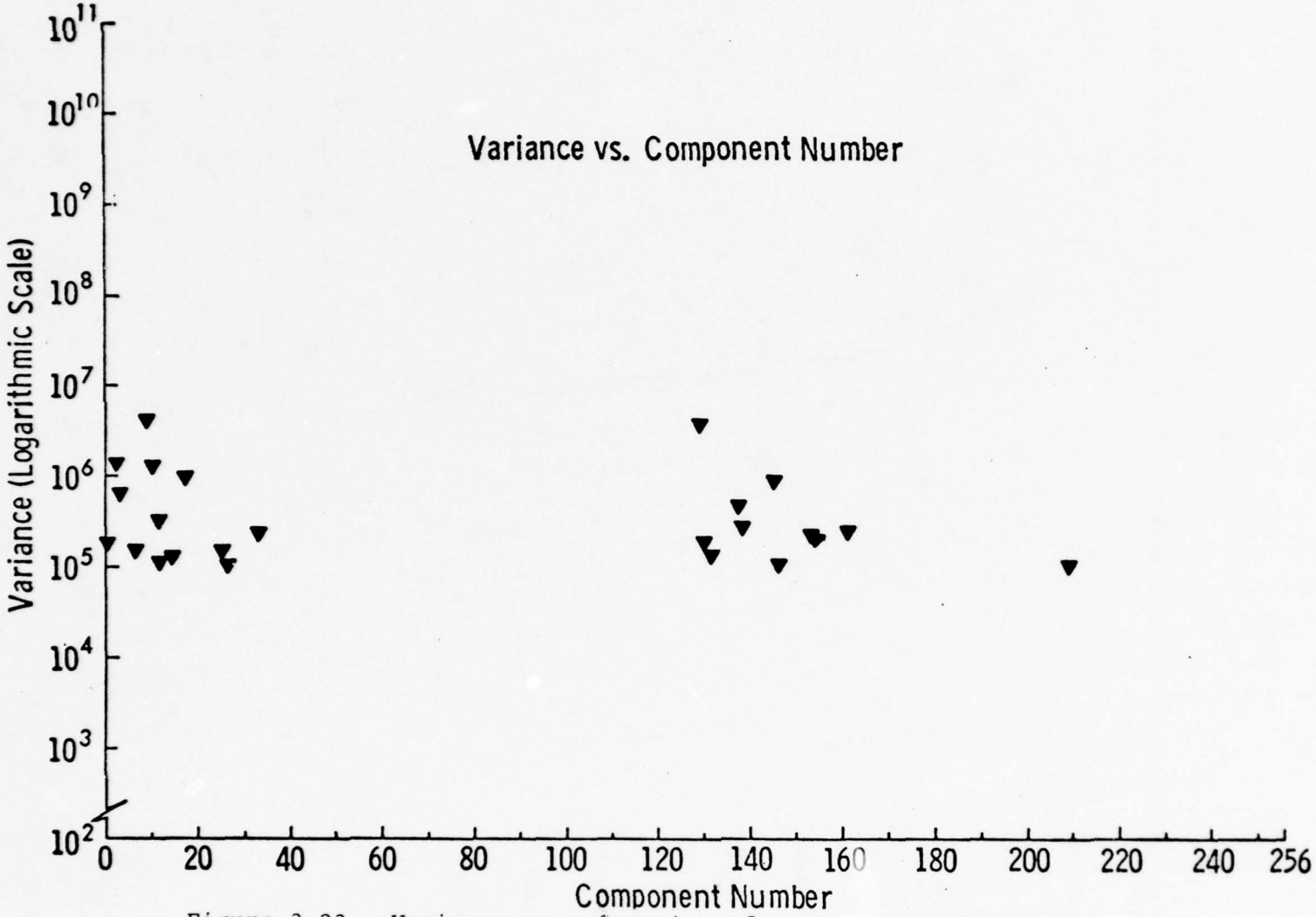


Figure 3.23. Variance as a function of component number for low and midrange components of the DLB Transform.

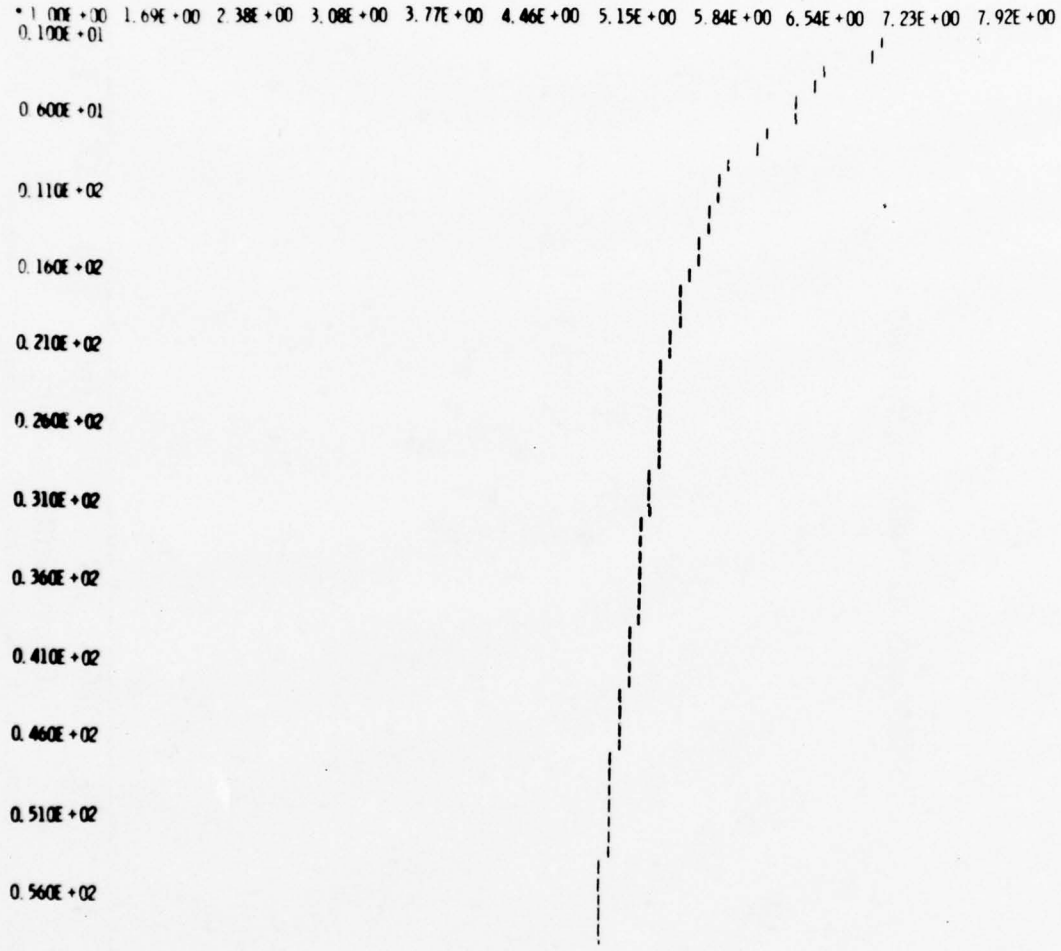


Figure 3.24. Sorted variance as a function of component number for DLB.

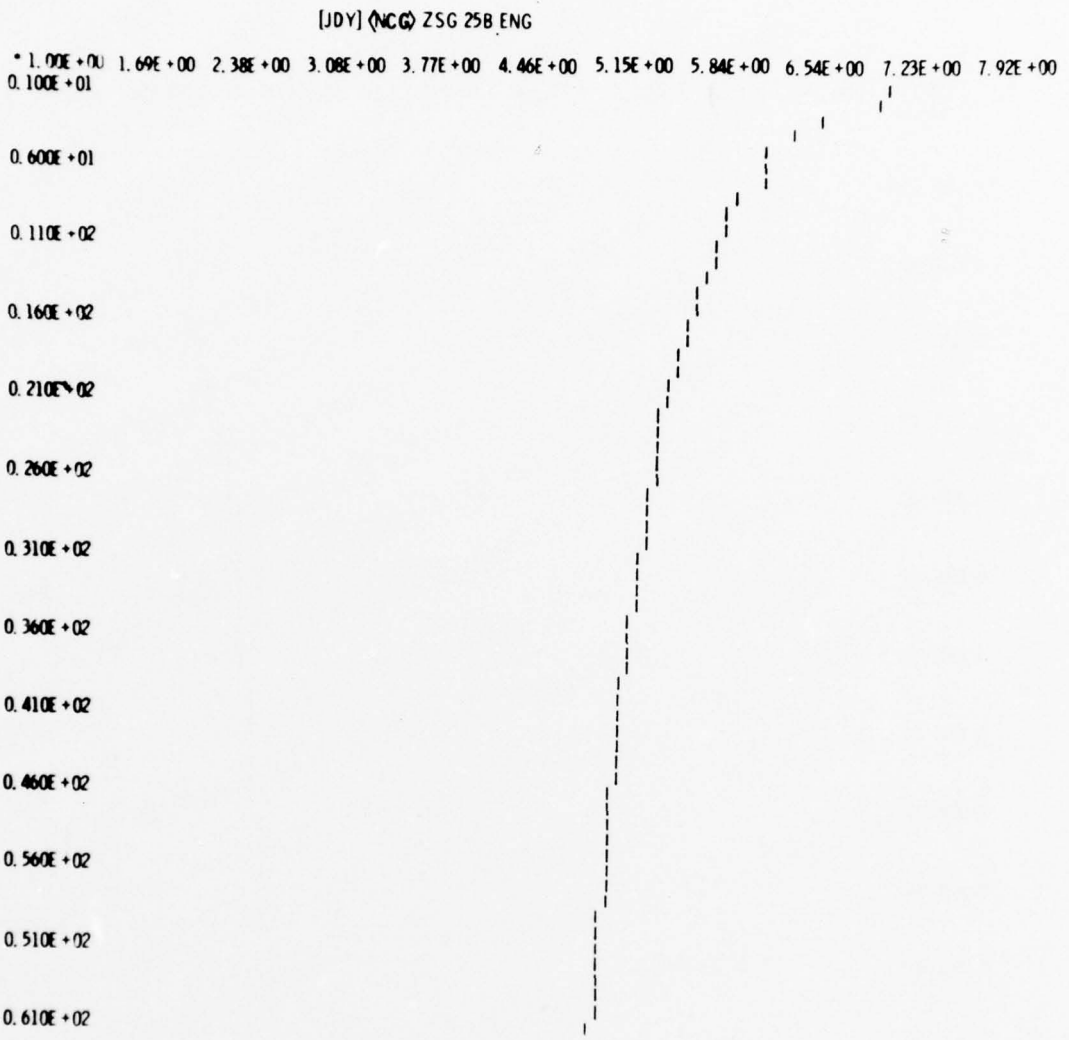


Figure 3.25. Sorted variance as a function of component number for Slant.

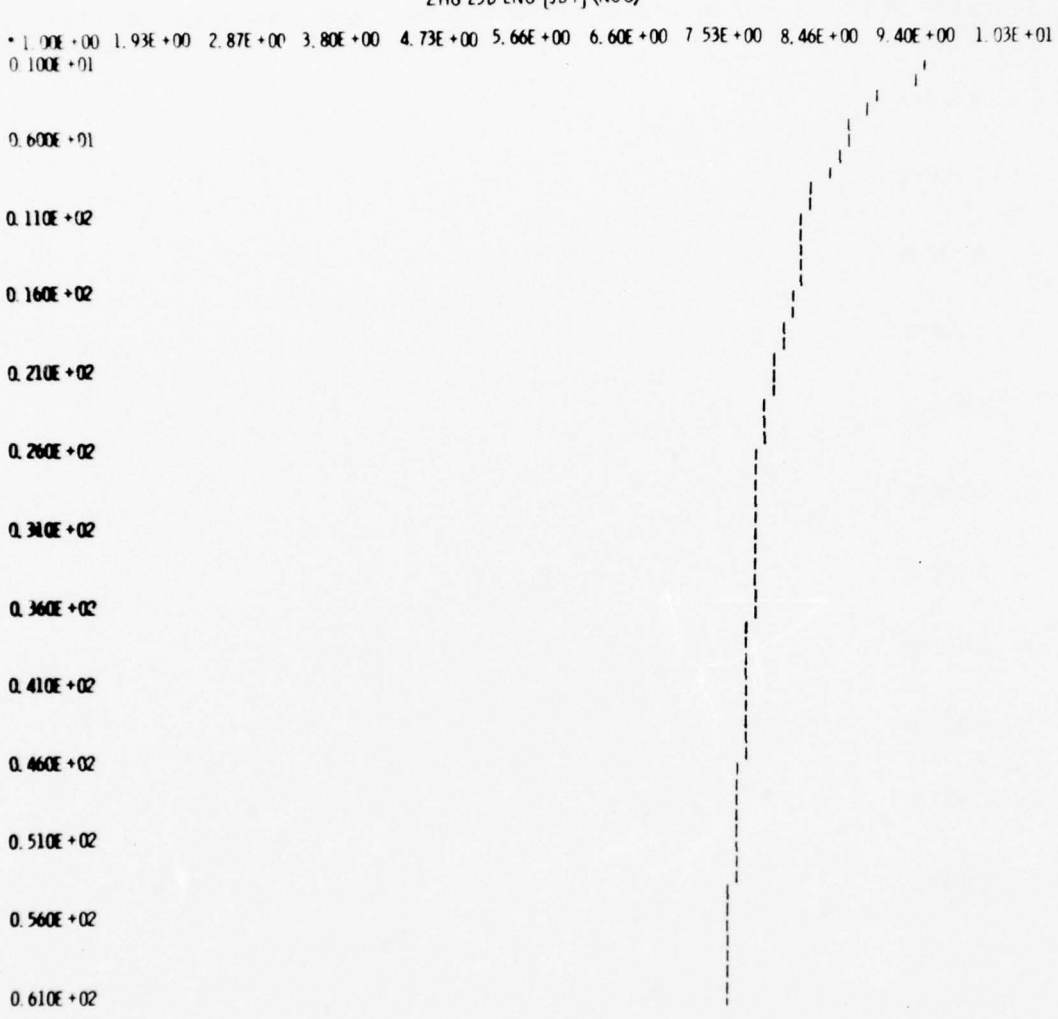


Figure 3.26. Sorted variance as a function of component number for Hadamard.

See Table 3.3. There is no question that the Hadamard is the poorest transform based on this criteria, followed by the SLANT and the DLB being the best of the three. The correlation values of the error image on the SLANT and DLB are less than the Hadamard. A low correlation means less dependence of neighboring cell in the error image, therefore less redundant information not capitalized upon. Stated simply the SLANT and DLB energy compression is more efficient in reducing redundancy. Typical error pictures for Scenes 1 and 2 (tank and truck on bridge) are shown in Figures 3.27 and 3.28. These pictures indicate that the errors are predominant around the edges of the target. The high spatial correlation is reflected in higher values of correlated RMS error, showing usefulness of the correlated RMS error as a measure of the magnitude and spatial distribution of errors.

As stated under theoretical background the correlation measure is dependent upon the distance used in comparing the probability of occurrence of any two gray levels. A few graphs of the correlation measure as a function of distance is given in Figures 3.29, 3.30 and 3.31. The reconstructed images are shown in Figures 3.32 and 3.33.

5. Optimum Bit Encoding

The bit encoding procedure must determine the optimum number of projections to use and the bits per projection. Therefore, given that a finite number of bits are to be distributed among a maximin number of projections (N OPT), the optimum number of projection will be determined as well as the bits per projection. The optimizing criteria is the overall mean squared error. The number of bits necessary for each projection is determined by the distribution and variance of that particular component taken over the total image. The actual value of the projection within this distribution is quantized by minimum variance quantization. [2]

TABLE 3.3

SCENE 1 8:1 ENERGY COMPRESSION

TRANSFORM	% ENERGY	RMS ERROR	CORRELATION	LAG DIST	CORRELATED RMS ERROR
HAD	99.84	.3264489E01	.95214E-01	1	.31082E00
			.93699E-01	2	.30588E00
			.91331E-01	5	.29814E00
			.88209E-01	10	.28796E00
SLANT	99.85	.31572E01	.92728E-01	1	.29276E00
			.92187E-01	2	.29105E00
			.89653E-01	5	.28305E00
DLB	99.01	.3112585E01	0.8236E-01	1	.25916E00
			0.8405E-0	2	.26163E00
			0.8253E-01	5	.25688E00
			0.7974E-01	10	.2482E00

SCENE 2 8:1 ENERGY COMPRESSION

TRANSFORM	% ENERGY	RMS ERROR	CORRELATION	LAG DIST	CORRELATED RMS ERROR
HAD	99.84	.46239E01	.116777E00	1	.53997E00
			.118379E00	2	.54738E00
			.116557E00	5	.53895E00
SLANT	99.85	.44744E01	.112432E00	1	.50306E00
			.115880E00	2	.51849E00
			.116129E00	5	.51961E00
			.114492E00	10	.51205E00
DLB	98.95	.44234E01	.9715E-01	1	.4297E00
			.1014E00	2	.4485E00
			.10469E00	5	.46308E00

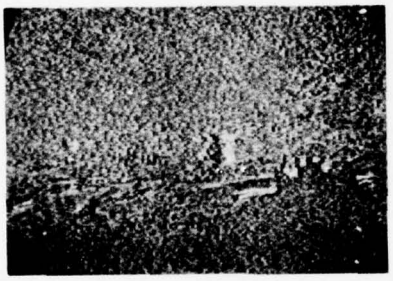


Figure 3.27. Error Image Scene 1 DLB

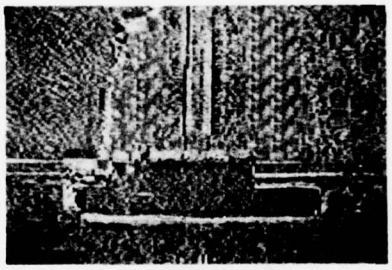


Figure 3.28. Error Image Scene 2 DLB

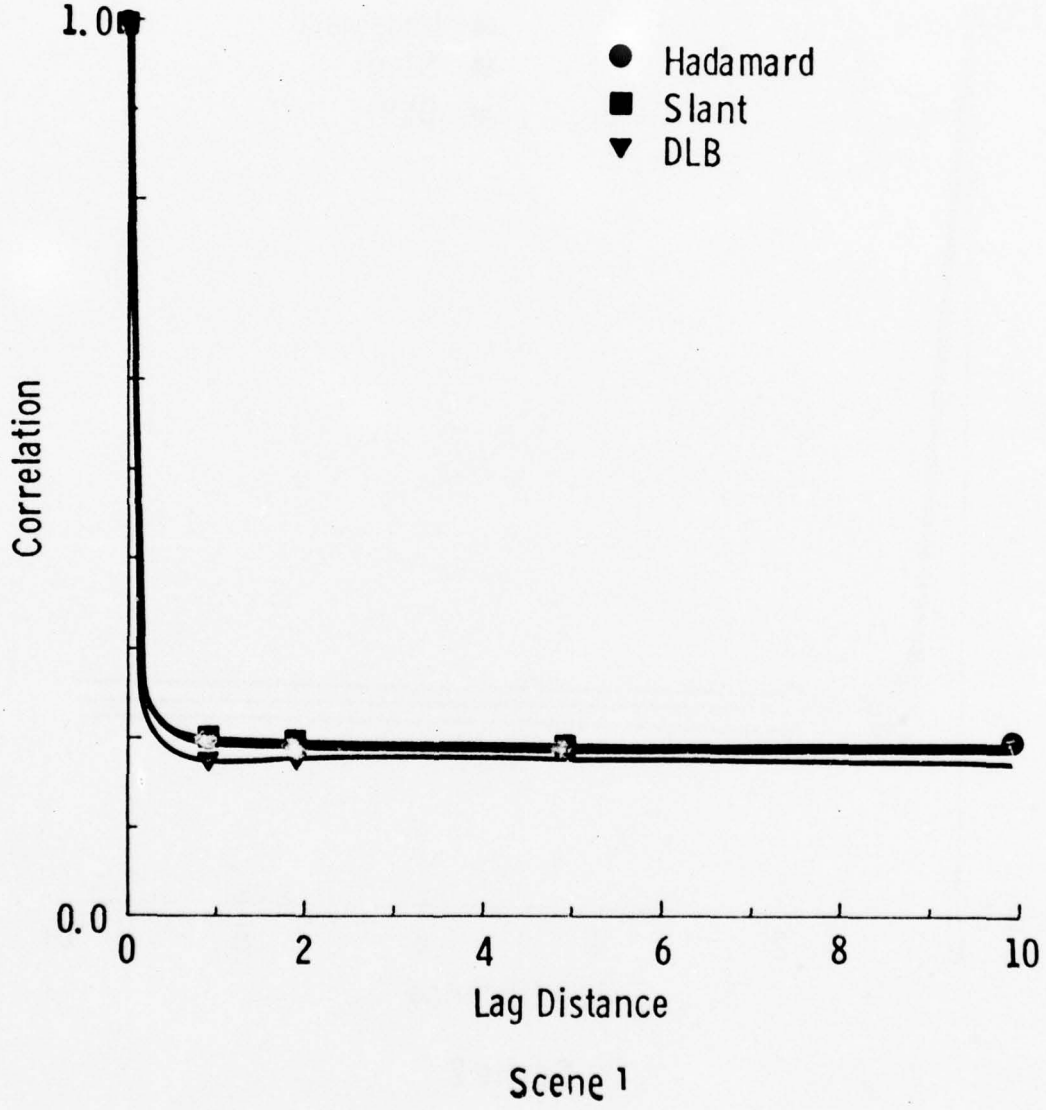


Figure 3.29. Correlation Function vs. Distance.

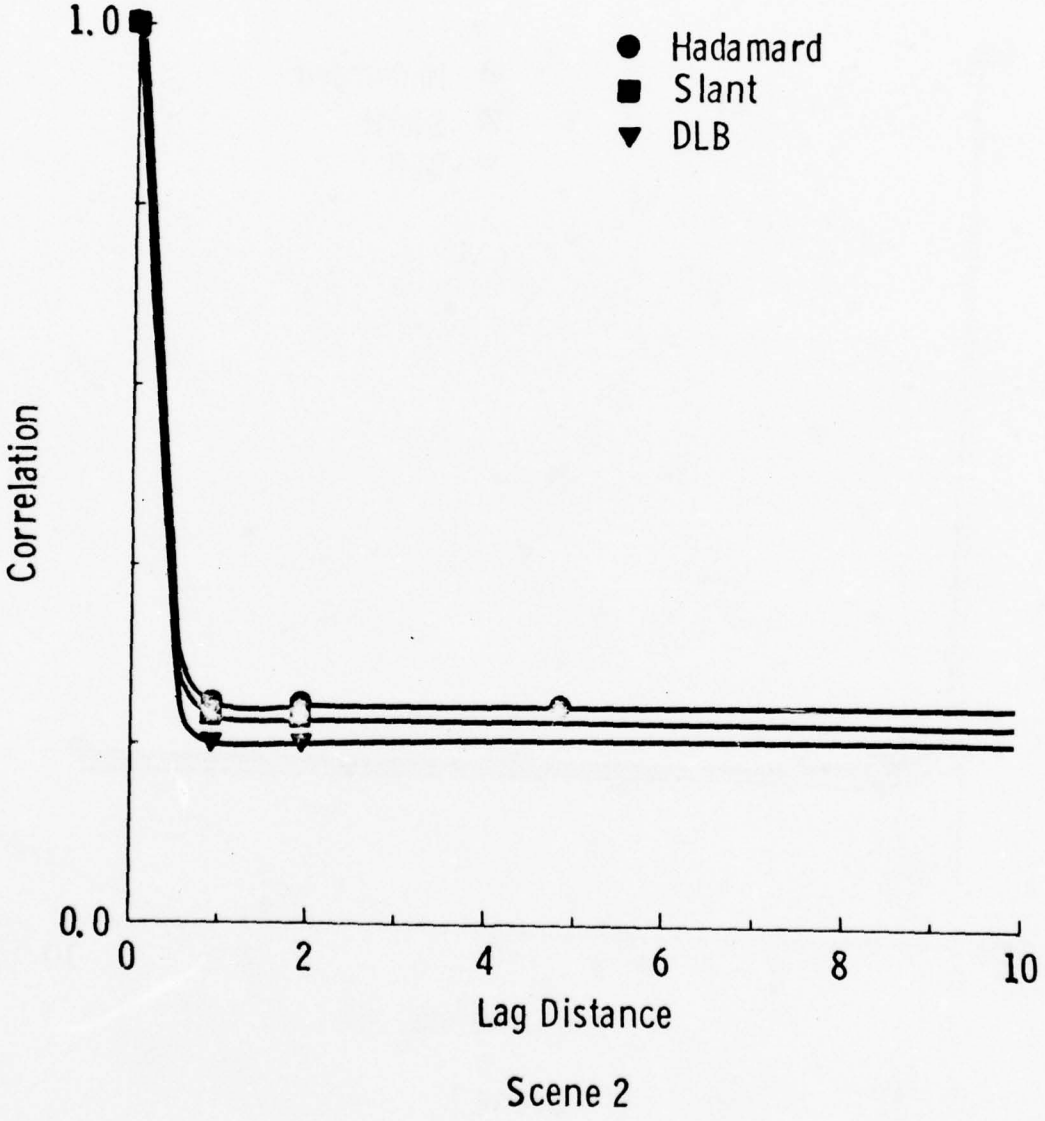


Figure 3.30. Correlation Function vs. Distance.

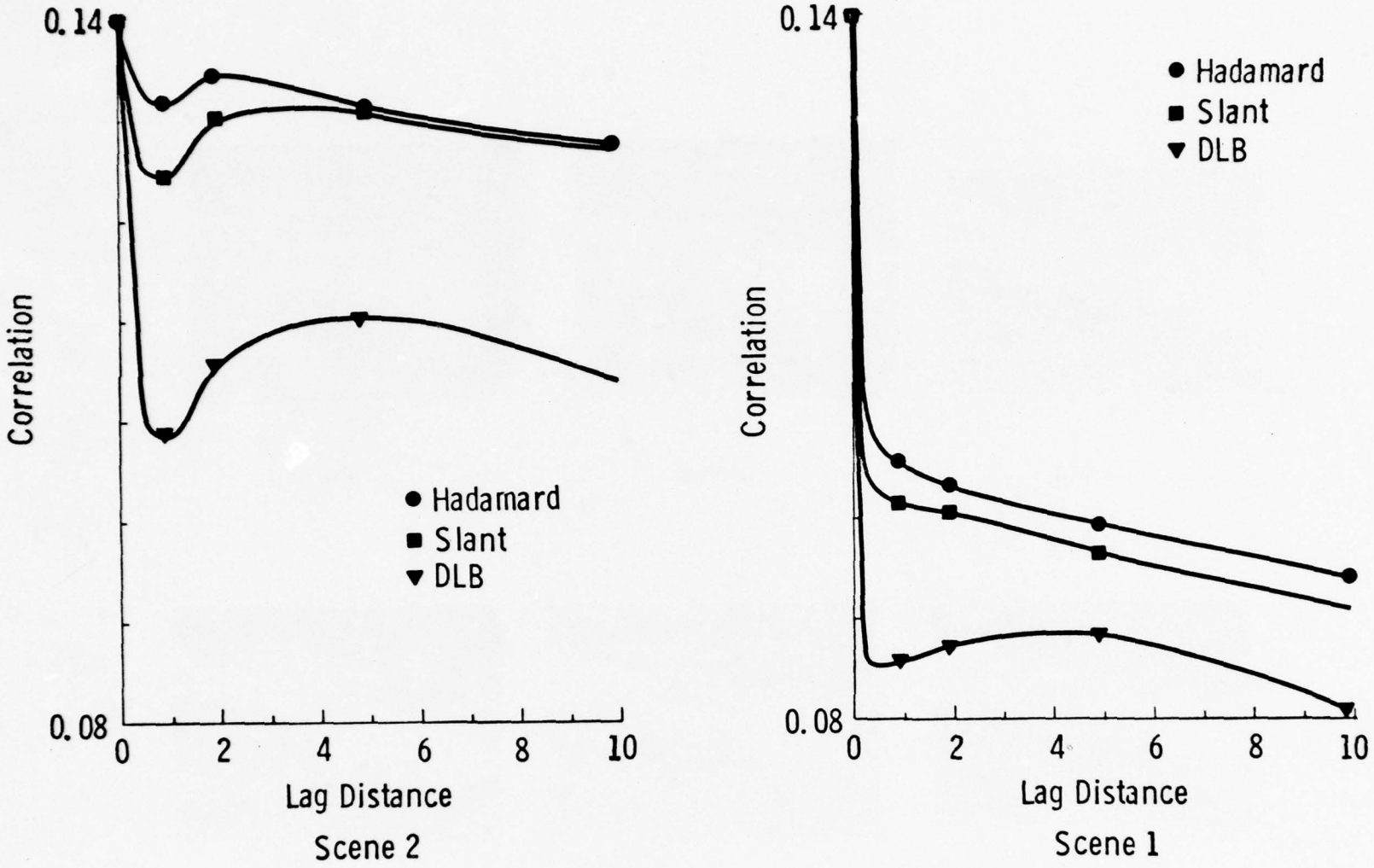


Figure 3.31. Correlation Function

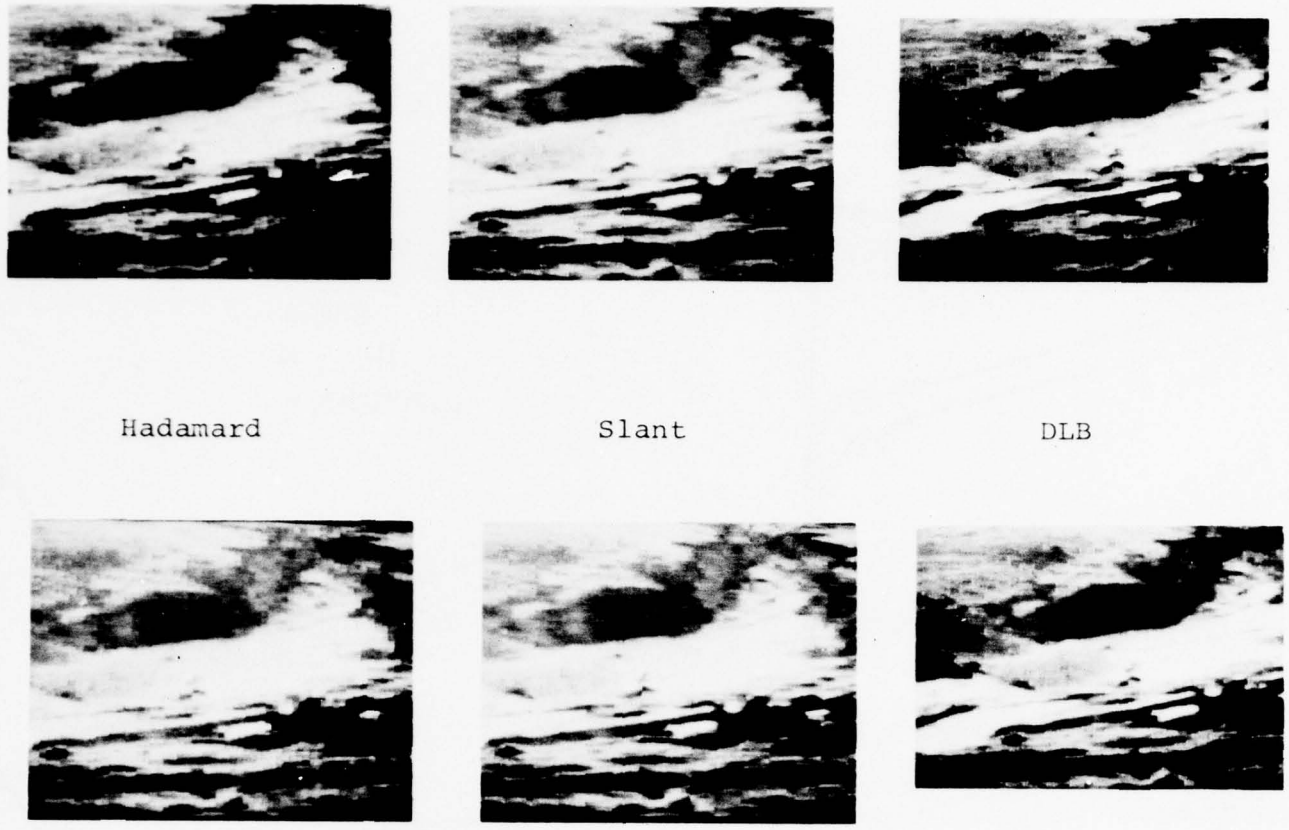
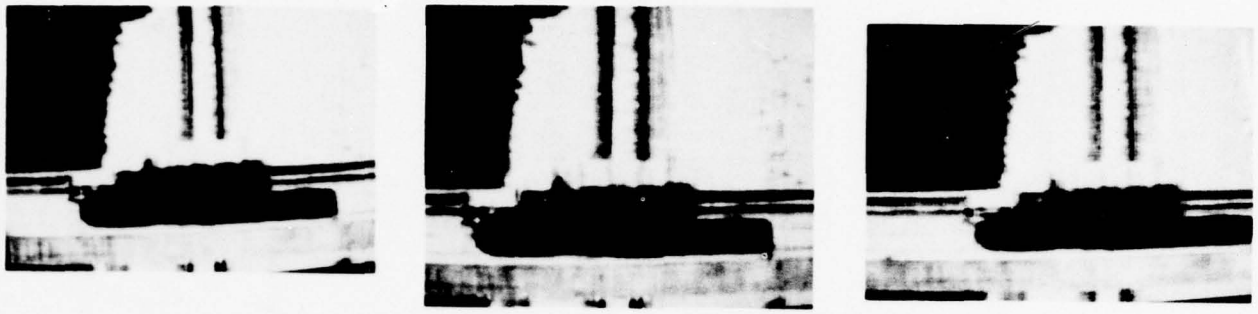
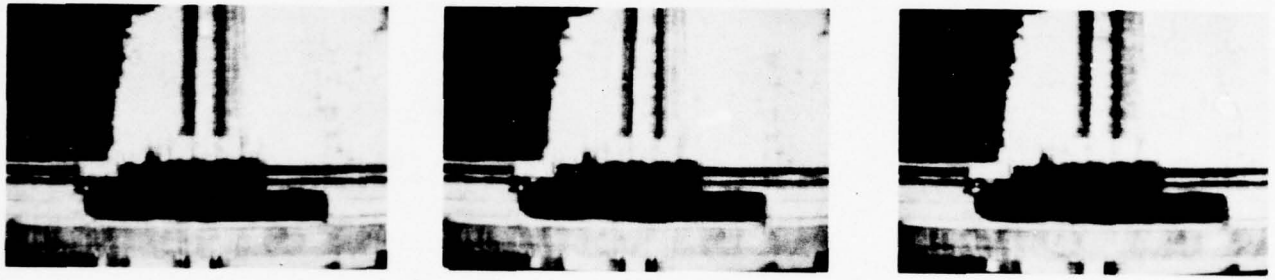


Figure 3.32. Energy Compression - Scene 1



8:1

Hadamard Slant DLB



10:1

a. Minimum Variance Quantization

Since a digital transmission system has a finite transmission rate a quantizer must sort the input signal into a finite number of ranges, N. The input signal is in general a floating point value representing a transformation of a data vector in the frequency or transformed domain. For a given N the system is described by specifying the end point, x_K, of the K^{th} input frequency component and y_K an output level corresponding to each input range. Of course, the input frequency components correspond to the number of projections retained in the transformation for transmission. Each component will have a distribution which indicates the variation of values that components may take on. The optimizing criteria is the minimum (overall) mean square error. This criteria may be utilized by defining the distortion of the quantization process as some statistic of the quantization error. The distrotion, D, may be defined as the expected value of $f(\varepsilon)$, where ε is the quantization error and $p(x)$ is the density function of the input amplitudes. The function $f(\varepsilon)$ is assumed to be a differentiable function.

Then $$D = E\{f(S_{in}-S_{out})\}$$

$$= \sum_{K=1}^{N} \int_{y_{K-1/2}}^{y_{K+1/2}} (x-y_K)^2 \, p(x)dx$$

where $x_{i+1} - x_i = y_{K+1/2} - y_{K+1/2} = \Delta y$, an input range.

The distortion in terms of each step is, therefore:

$$\overline{(x-y_K)^2} = \int_{y_{K-1/2}}^{y_{K+1/2}} (x-y_K)^2 \, p(x)dx = \sigma_K^2$$

where $p(x)$ is considered a constant over range of integration and equal to:

$$p(y\ avg) = \frac{y_{K+1/2} - y_{K-1/2}}{2} \quad \text{for intervals spaced sufficiently close.}$$

Then:

$$\sigma_K^2 = \frac{p(y_{avg})}{3} \left[(y_{K+1/2} - y_K)^3 + (y_K - y_{K-1/2})^3 \right] = 0$$

$$\frac{\partial(\sigma_K^2)}{\partial y_K} = p(y_{avg}) \left[-(y_{K+1/2} - y_K)^2 + (y_K - y_{K-1/2})^2 \right] = 0$$

$$\text{or } y_K = \frac{y_{K+1/2} + y_{K-1/2}}{2}$$

thus the condition for minimizing σ_K is to locate the output value y_K halfway between $y_{K+1/2}$ and $y_{K-1/2}$.

Therefore:

$$y_{K+1/2} = y_K + \frac{\Delta y_K}{2}$$

$$y_{K-1/2} = y_K - \frac{\Delta y_K}{2}$$

Substituting into above:

$$\sigma_K^2 = \frac{(\Delta y_K)^3}{12} \quad p(y_K)$$

The total mean square error voltage is found by summing over all levels (total distortion):

$$D = \frac{1}{12} \sum_{-N}^{N} p(y_K)(\Delta y_K)^3$$

As given by the paper of Joel Max the distortion given the P^{th} component tangent at a point is given empirically as: [2]

$$D = \sigma_p^2 \, N^{-q}$$

N is the number of levels

σ_p^2 variance of P^{th} components

D total distortion for N levels

q a constant to be determined

The distortion or total mean squared error of any frequency component can be measured as a function of the number of bits allocated or number of levels. The range of the component which is divided into these levels is determined by the variance of the frequency component. A plot of distortion as a function of these levels for several frequency components is shown in Figure 3.34. From these plots it is apparent that large variance (high energy) components must be divided into a higher number of levels to minimize the distortion.

In our program this quantization error is given by:

$$Q(I) = \sigma_I^2 \; C^{-2*Nbits(I)}$$

where I is the I^{th} projection.

c is 1.78 for normal distribution and

Nbits is the number of bits assigned to the I^{th} component [2].

The truncation error due to using p projections out of a total number of NPR projections is given by:

$$T(p) = \sum_{i=1}^{NPR} VAR(I) - \sum_{i=1}^{P} VAR(I)$$

$$= L - \sum_{i=1}^{P} VAR(I)$$

where VAR(I) is the variance of the I^{th} component.

The total error is given by [2]:

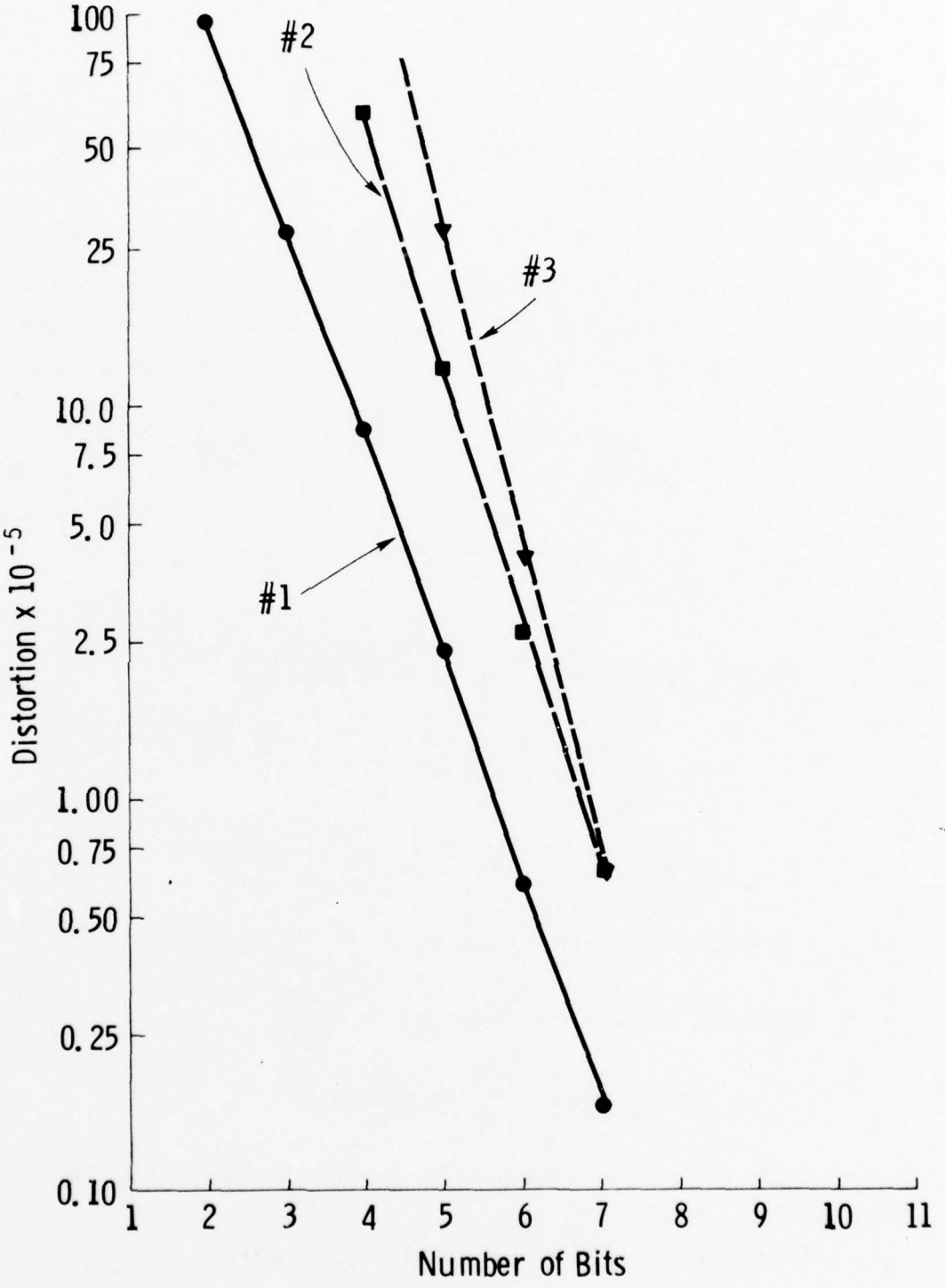

100
75
50
25
10.0
7.5
5.0
2.5
1.00
0.75
0.50
0.25
0.10
Distortion x 10⁻⁵
#1
#2
#3
1
2
3
4
5
6
7
8
9
10
11
Number of Bits

Figure 3.34

$$\text{Total Error} = L - \sum_{i=1}^{p} \text{VAR}(I) + \sum_{i=1}^{p} \text{VAR}(I)\, C^{-2\text{Nbits}(I)}$$

Truncation error min Var. quantization error

An approximate solution to the problem posed above can be obtained by treating the discrete variables in the above as continuous variables. [3] The value of NOPT is given by:

$$\text{NOPT} = \min_{1 \le p \le \text{NPR}} \left\{ \text{VAR}(p) - \text{CON}^{1/p} * [D(p)] \right\},$$

where:

$$\text{CON} = \prod_{I=1}^{p} \text{VAR}(I) / C^{2*\text{MBITS}}$$

and:

$$D(p) = 1.0 + \log_e \text{VAR}(p) + \frac{2*A*\text{MBITS}}{p} - \frac{1}{p} \sum_{I=1}^{p} \log_e \text{VAR}(I)$$

and:

$$A = \log_e C$$

The bit assignments are given by:

$$\text{NBITS}(J) = 0.5*\text{Log}_e(\text{VAR}(J)) + \frac{\text{MBITS}}{\text{NOPT}} - \frac{1}{2x\text{NOPT}} \sum_{I=1}^{\text{NOPT}} \text{Log}_e(\text{VAR}(I))$$

Since NBITS(I) was treated as a continuous variable above, $\sum_{I=1}^{\text{NOPT}} \text{NBITS}(J) = \text{MBITS}$ will not be true after we round off each NBITS(I) to an integer. To make this equality hold, we need to remove or add 1 bit at a time to the array NBITS. See Appendix II.

The value of C = 1.78 used for normal distribution must be verified for this process. If the distortion as a function of number of levels is plotted on a log-log graph then the tangent at a point will be the slope of the curve at that point. This is the value of q in the equation:

$$D = \sigma_p^2 \; N^{-q}$$

then by taking the log of both sides we have:

$$\log D = \log \sigma_p^2 - q \log N$$

or

$$-q = \frac{\log D}{\log N} - \frac{\log \sigma_p^2}{\log N}$$

The quantization error given for our program is:

$$Q(I) = \sigma^2(I) \; C^{-2 \; Nbits}$$

therefore $N^{-q} = C^{-2 \; Nbits}$ if the variances of these expressions are equal. Then:

$$C = \text{Anti-log} \; \frac{q \log N}{2 \; Nbits}$$

In our optimization process, however, the total distortion is not only based on the quantization error but also the truncation error. This allows the (Nbits), number of bits, to change for any component variance. Therefore, C must be related to how close the distribution is to a normal distribution for a given variance.

A further test to determine how close the frequency distribution of a component is to a normal distribution can be achieved by the Mann-Whitney-Wilcox text. [3]
Let X and Y be stochastically independent random variables of the continuous type.
Let F(x) and G(y) denote the distribution functions of X and Y.
Let $X_1, X_2, \ldots, X_m$ and $Y_1, Y_2, \ldots, Y_n$ denote independent

samples from these distributions. Define:

$$Z_{ij} = 1 \qquad X_i < Y_j$$
$$= 0 \qquad X_i > Y_j$$

and consider the statistic

$$u = \sum_{j=1}^{n} \sum_{i=1}^{m} Z_{ij}$$

and note that

$$\sum_{i=1}^{m} Z_{ij}$$

counts the number of values of X which are less than Y_j, $Y_j, j=1,2,\ldots,n$ thus U is the sum of these n counts. The smallest value that U can take on is 0 and the largest value that U can take on is mn. Thus the space of U is $\{u: \mu=0, 1, 2,\ldots,mn\}$. If U is large the values of Y tend to be larger than the values of X. Thus if we test the hypothesis that H_0: $F(z) = G(z)$ against

H_1: $F(z) \geq G(z)$

the critical region is $U \geq C$.

To determine the size of the critical region we need the distribution of U when H_0 is true.

It has been proven that: [4]

$$\frac{U - \frac{mn}{2}}{\frac{mn(m+n+1}{12}} \quad \text{is } N(0,1)$$

therefore, this hypothesis can be tested for various significant levels. U may be determined by:

$$U = T - \frac{n(n+1)}{2}$$

where T is the sum of the ranks of $Y_1, Y_2,\ldots,Y_n$ among the m+n items $X_1,\ldots,X_m, Y_1,\ldots,Y_n$ once this combined sample is ordered.

Therefore, at the α significance level we have:

$$P_r \left[\frac{U - \frac{mn}{2}}{\sqrt{\frac{mn(m+n+1)}{12}}} \right] \geq N(x) = \alpha$$

$N(x)$ table value for $(1-\alpha)$ of normal distribution.

$$\text{If } Pr \left[U \geq \sqrt{\frac{mn(m+n+1)}{12}} \quad N(x) + \frac{mn}{2} \right] = \alpha$$

The histogram of these components is shown in Figures 3.35, 3.36 and 3.37. The question which must be answered is: are the frequency components sufficiently close to a normal distribution to allow the value of C = 1.78 in the optimum bit allocation program to be used? The results of this test of hypothesis indicates that all A-C frequency components are sufficiently close to normal to use the value C = 1.78 in the quantization process. The d.c. component is neither normal nor uniform although it tends to appear to be a uniform density function. For this reason a value of C = 1.78 in the minimization process was used.

The value of C = 5.61 and C = 1.78 was compared using the bit allocation program to show the effect of changing this value from the normal assumption. A Hadamard transform with compression of 1 bit/pel was used. See Figure 3.38. The fact that the image with a C value of 1.78 is much better indicates agreement with the hypothesis test.

Reconstructed images of the Hadamard, SLANT, and DLB using this optimum bit allocation for Scene 1 (tank in ditch) and Scene 2 (truck on bridge) are shown in Figures 3.39, 3.40 and 3.41. Visually these reconstructed images can be compared to component compression ratios. The 1 bit/pel is equivalent to 6:1 component compression, while the .75 bits/pel and .5 bits/pel are equivalent to 8:1 and 12:1 component compression ratios, respectively. Visual comparison of the reconstructed image shows that .5 bits/pel

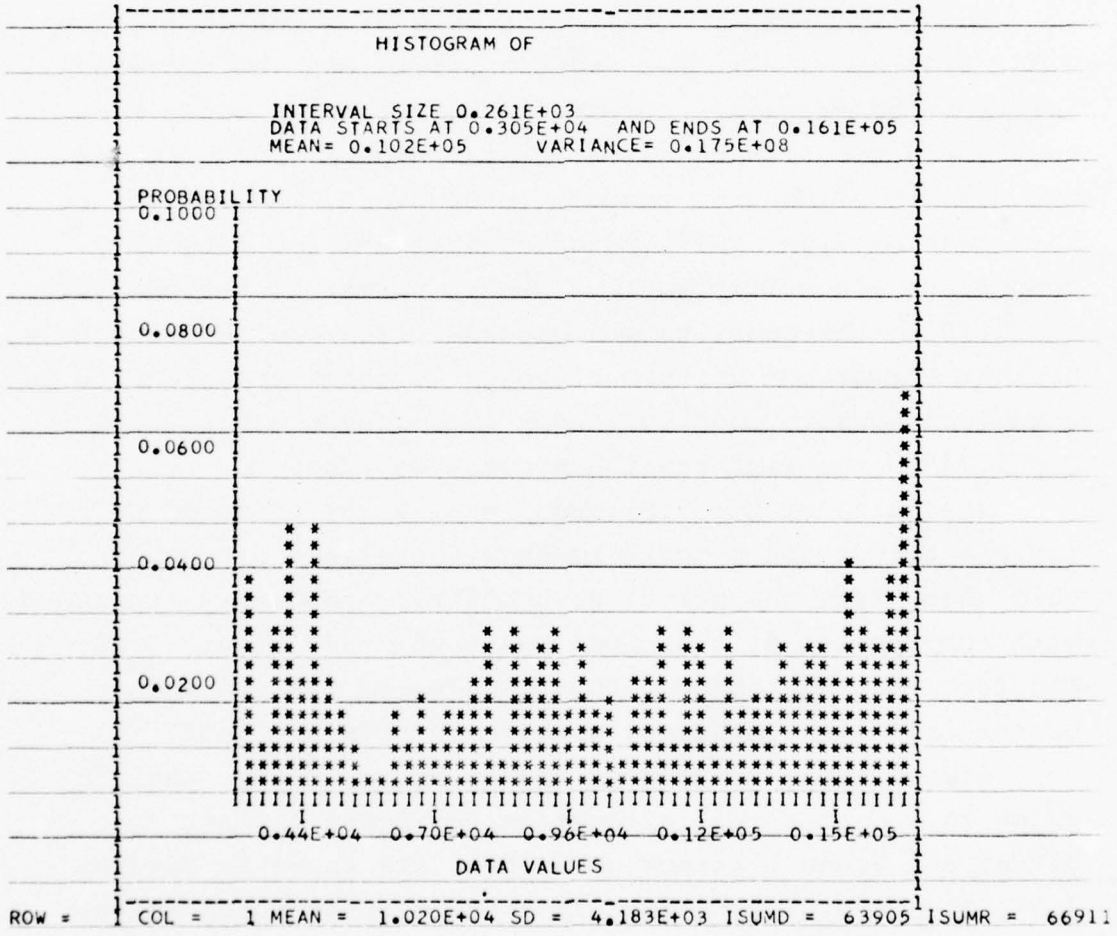


Figure 3.35. Distribution of Components

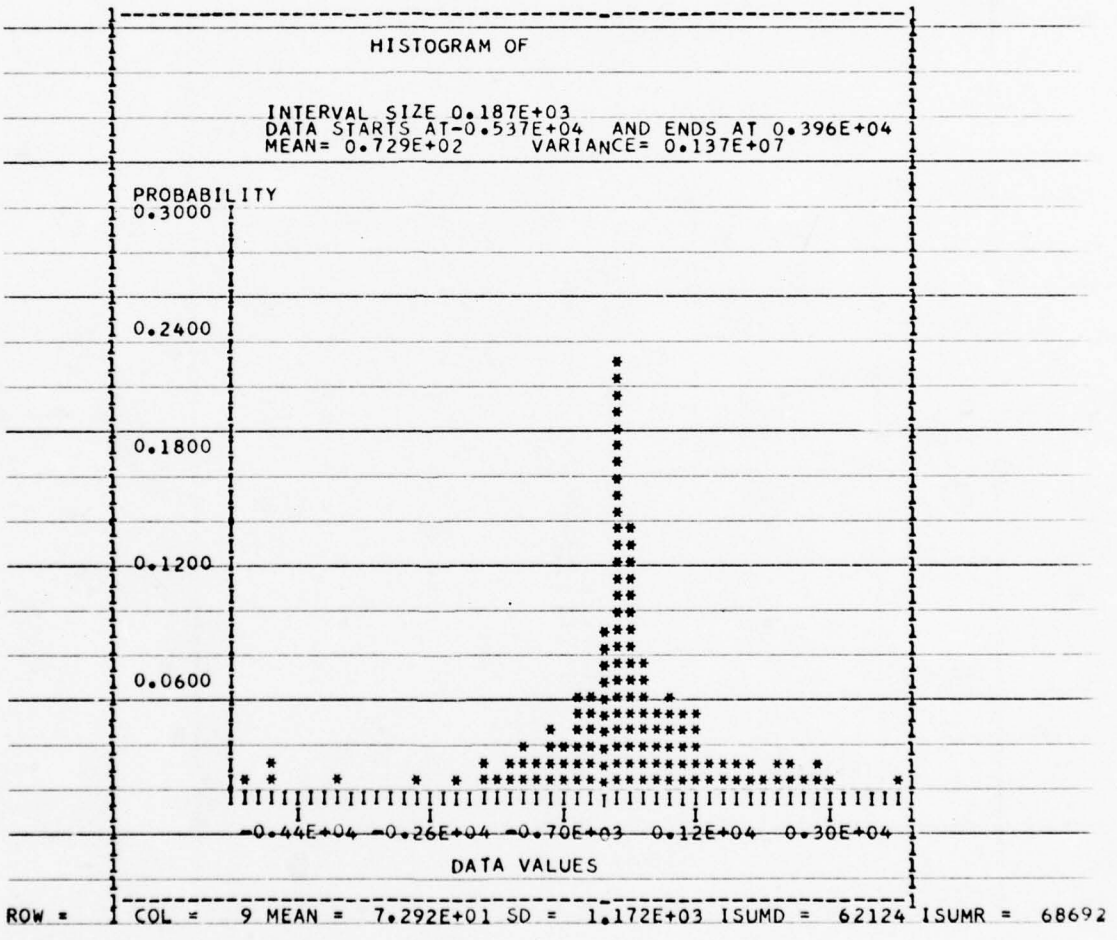
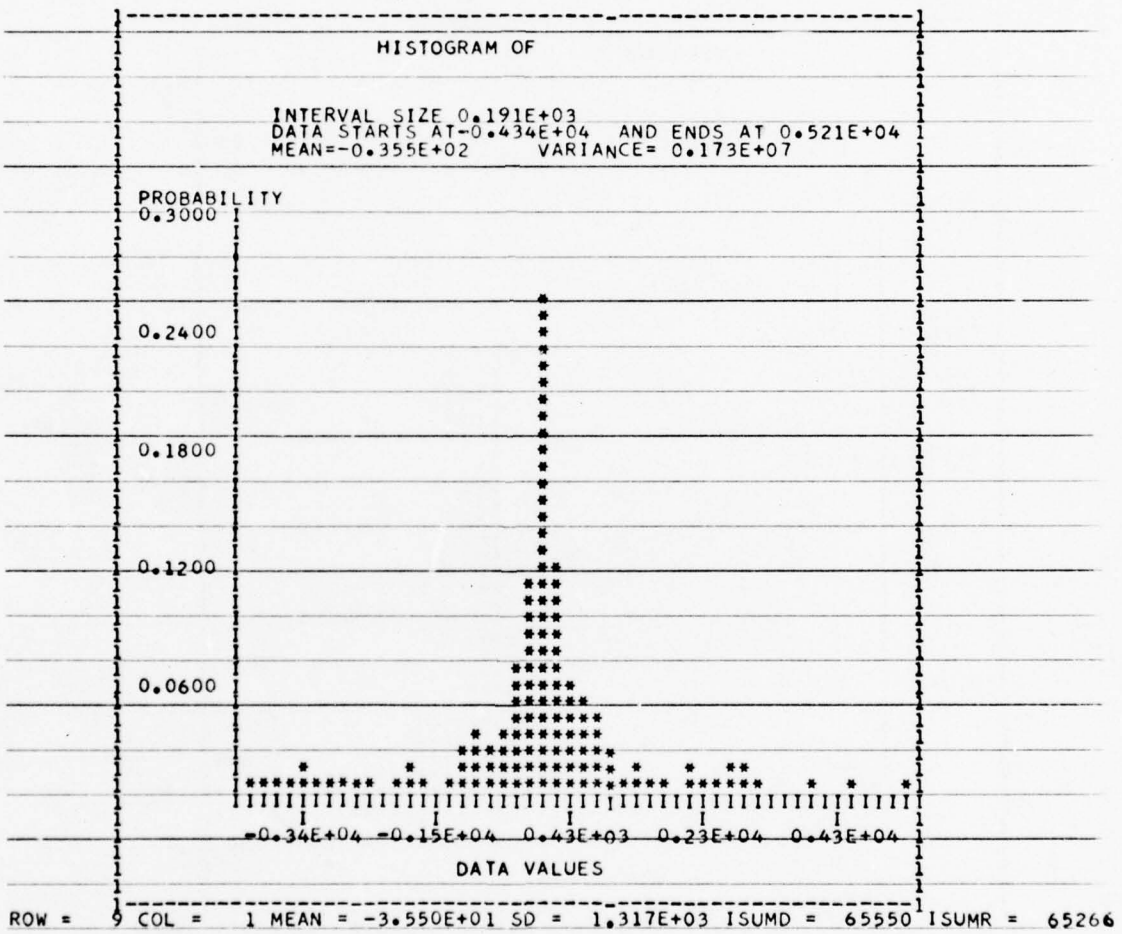
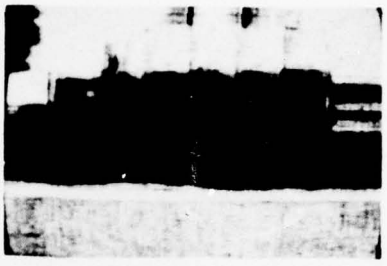


Figure 3.36. Distribution of Components

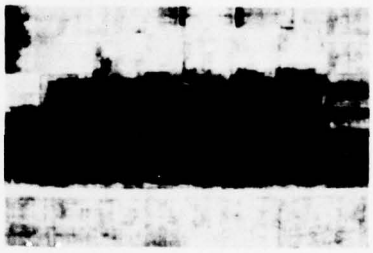


ROW = 9 COL = 1 MEAN = -3.550E+01 SD = 1.317E+03 ISUMD = 65550 ISUMR = 65266

Figure 3.37. Distribution of Components



C = 1.78



C = 5.61

Figure 3.38. Visual Comparison of Two Values of C

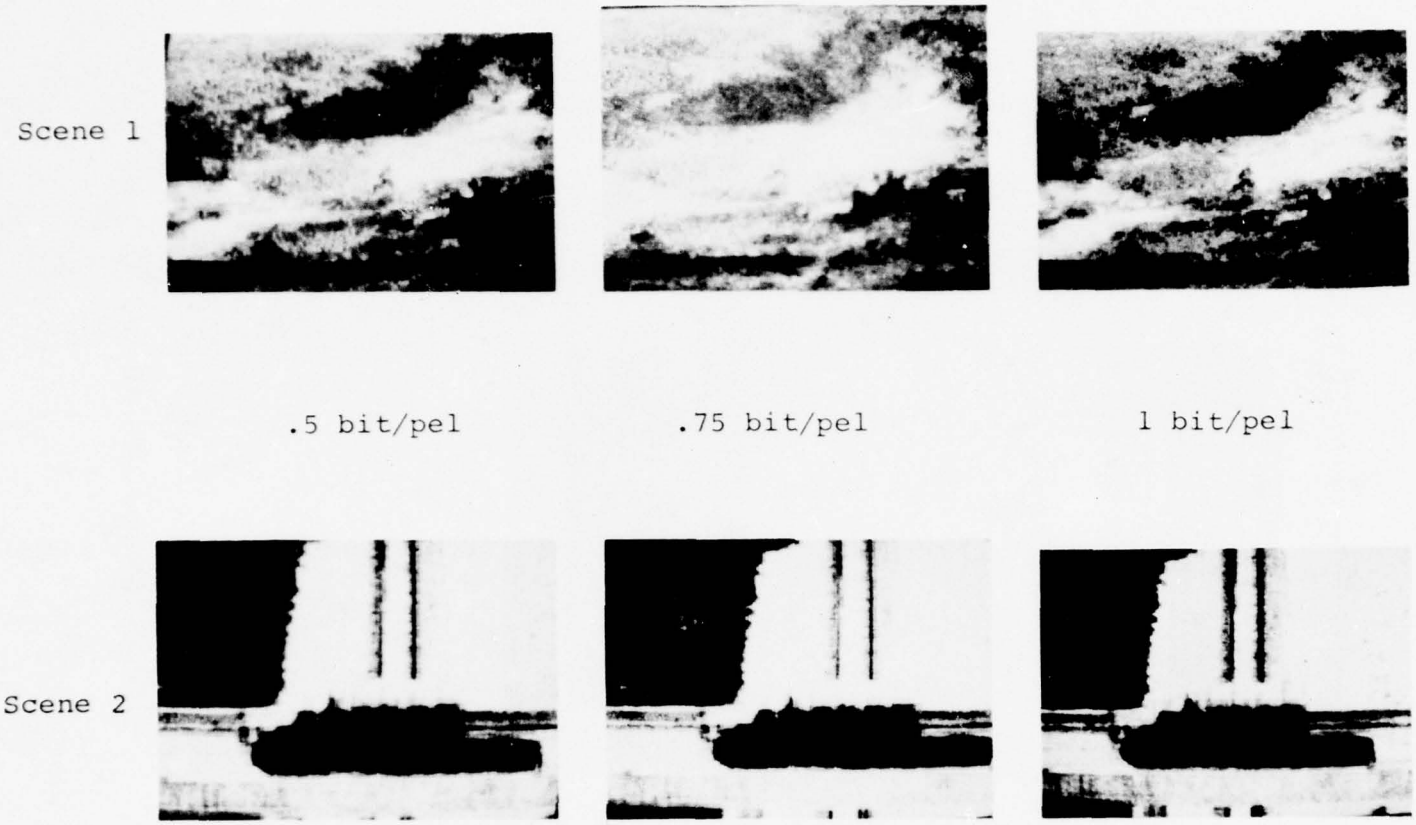


Figure 3.39. Hadamard Optimum Bit Allocation

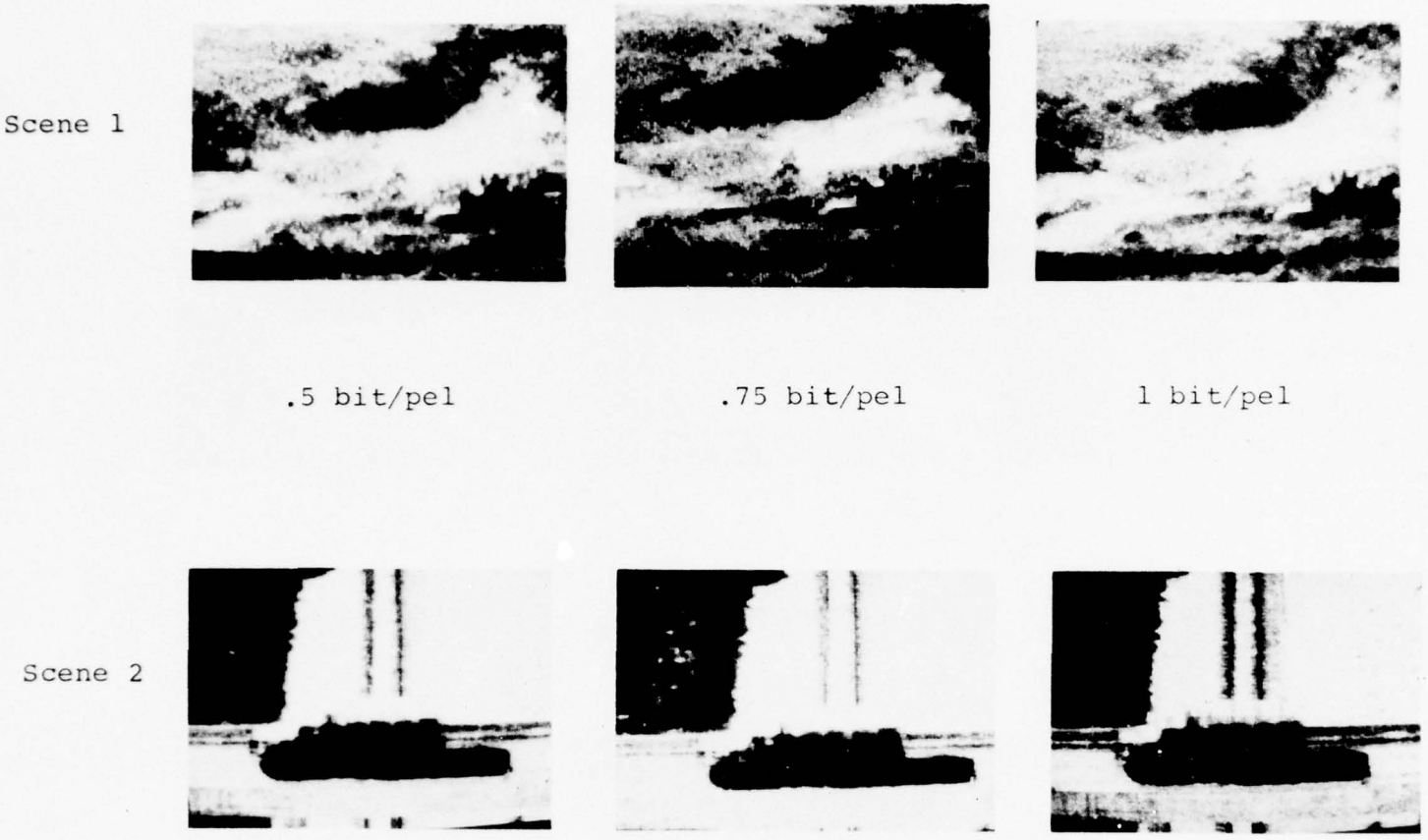


Figure 3.40. Slant Optimum Bit Allocation

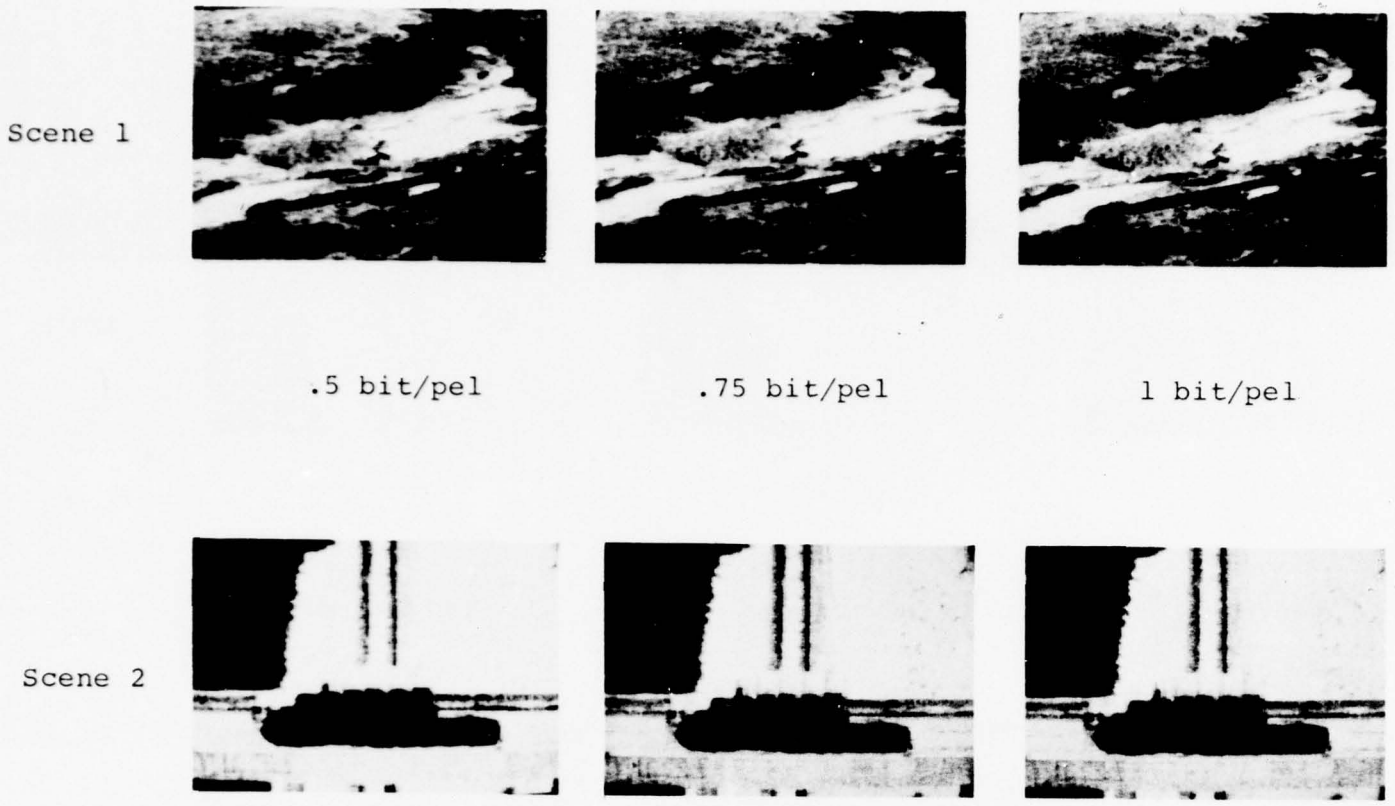


Figure 3.41. DLB Optimum Bit Allocation

is feasible with either the SLANT or DLB for a 512 x 512 image. The Hadamard has an increase in blocking at .5 bits/pel.

Using the variable word length coding scheme is more efficient in using less bits for components with less energy contribution and more bits for components with high energy contribution. The quality of the image (cosmetic effects) is better than the equivalent component compression. Or stated another way, for the same RMS error an additional 1.25-1.5 compression can be gained by optimum bit allocation.

In terms of the error criteria the RMS and RMS correlated errors are less with the variable length code as may be seen in Table 3.4 and Figure 3.42 for the same component compression. It was, therefore, determined that the bit encoding process would be used for evaluating the "best" transform and the interframe process.

SCENE 2

TRANSFORM	VISUAL COMMENT	RMS ERROR COMPRESSION RATIO 1.0 bit/pel.	.75 bit/pel.	.5 bit/pel.	CORRELATION AT .5 bit/pel.	LAG DIST	CORRELATED RMS ERROR
HADAMARD	Poor			5.34749	.123556	1	.660719
					.124702	2	.66684
					.122632	5	.65577
SLANT	Acceptable			5.1588	.121568	1	.62629
					.123536	2	.63643
DLB	Acceptable	4.0613	4.3679	4.8483	.11855	1	.57477
					.12081	2	.58572
					.11502	5	.55765

SCENE 1

TRANSFORM	VISUAL COMMENT	RMS ERROR COMPRESSION RATIO 1.0 bit/pel.	.75 bit/pel.	.5 bit/pel.	CORRELATION AT 5 bit/pel.	LAG DIST	CORRELATED RMS ERROR
HADAMARD	Poor						
SLANT	Acceptable			4.9011	.114483	1	.561096
					.118413	2	.580359
DLB	Good	2.7664	2.9878	3.31762	.10329	1	.34268
					.10321	2	.34241
					.10225	5	.33923

TABLE 3.4

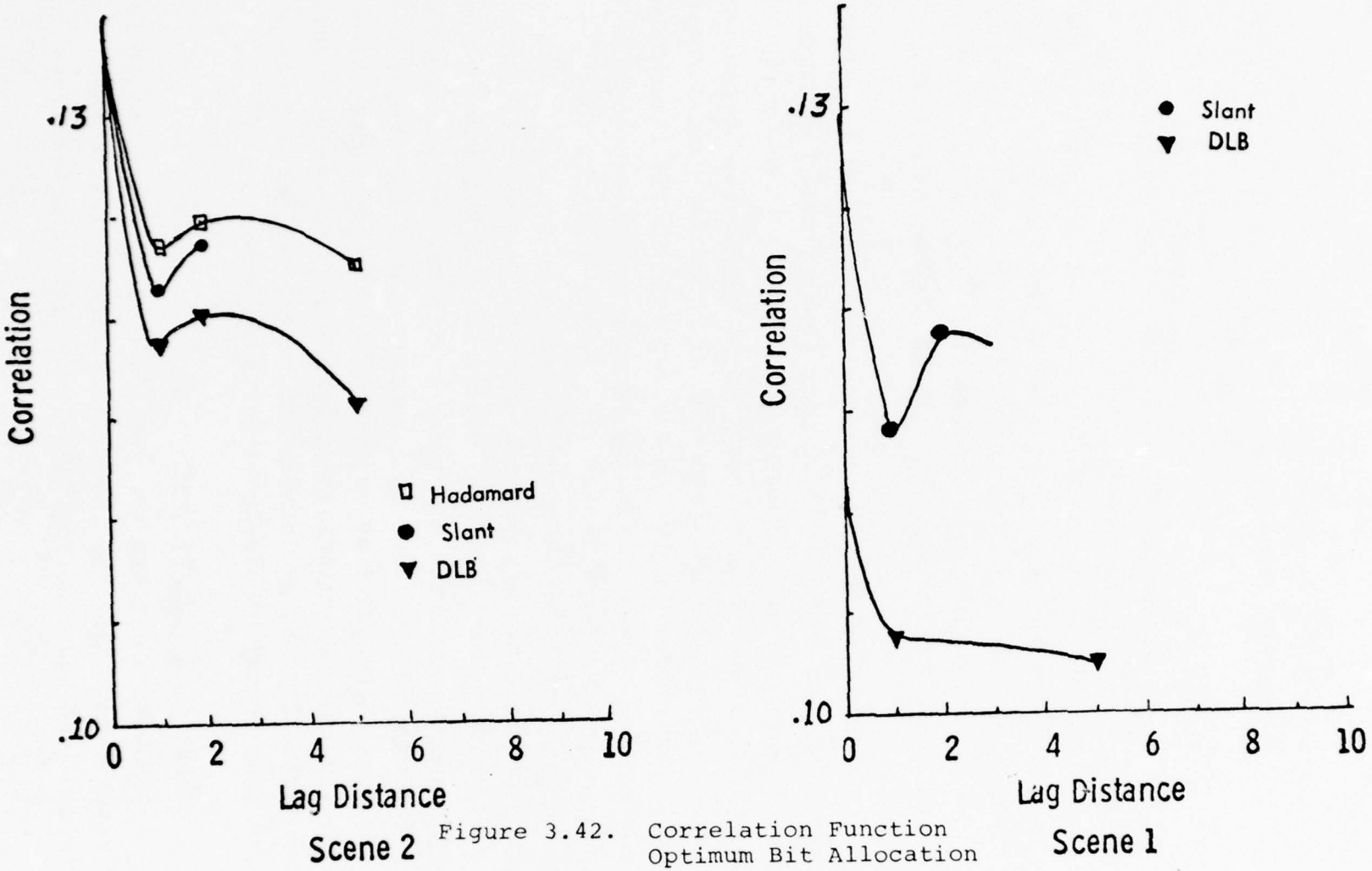


Figure 3.42. Correlation Function Optimum Bit Allocation

SECTION IV

A CRITICAL COMPARISON OF FAST TRANSFORMS

1. INTRODUCTION

In the previous chapters, we have examined the error criteria, the optimum transforms, and utilizing a few of the leading transforms, seen results of fixed and adaptive compression schemes. In this chapter we must examine the performance of each transform in the light of our chosen error criteria and bit allocation algorithm. The choice of the "best" transform may then be utilized in the interframe process.

In the search for the "best" fast transform other researchers have compared their most popular transform with a few of the other fast transforms. However, in the rapid growth of dimensionality reduction as applied to images, a comparison of all fast transforms at this time has not been made. The comparisons which are reported in the literature must also be viewed critically because each researcher is experimenting with different size images digitzed to 10, 8, and 6 bits utilizing different compression schemes. The error criteria has also been a variable in the past. Some researchers use only visual, others use RMS but the scaling of this error criteria has made it difficult to compare the performance of different researchers. It is therefore, the intent of this investiation to utilize the identical coding algorithm, error criteria, and representative image to reduce the variability to a minimum, and report the performance of each fast transform available for image compression.

2. Translation by the Mean

Prior to examining the results of all transforms it is desirable to modify the image input data to insure the minimum mean square error is achieved for each transform

process. This is accomplished by translating the vectors representing a subimage by the mean, of all vectors in that image. Returning to the principal components we can see why this is so.

Define P such that $P : R^N \to R^N$ and is a real N x N matrix. Then P is an orthogonal projection operator if and only if:

$$P^2 = P$$

$$P = P'$$

Then by the following theorem, the translation by the mean, in conjunction with principal components minimizes the error (see Haralick [1]).

Theorem 2: Let $x_1, x_2, \ldots, x_k$ be given vectors in R^N. Let $P : R^N \to R^N$ be an orthogonal projection operator onto the subspace V_M spanned by M eigenvectors of

$$\sum_{k=1}^{K} x_k x'_k \quad \text{with largest eigenvalues.}$$

Let $\varepsilon_1^2 = \sum_{k=1}^{K} (s_k - Px_k)'(x_k - Px_k)$

Let the vector Z be the arithmatic mean of $x_1, x_2, \ldots, x_k$:

$$Z = \frac{1}{K} \sum_{k=1}^{K} x_k$$

Also let P^*: $R^N \to R^N$ be an orthogonal projection operator onto the subspace V^*_m spanned by the M eigenvectors of

$$\sum_{k=1}^{K} (x_k - Z)(x_k - Z)' \quad \text{with largest eigenvalues, and}$$

then let $\varepsilon_2^2 = \sum_{k=1}^{K} ((x_k - Z) - P^*(x_k - Z))'$

$$((x_k - Z) - P^*(x_k - Z))$$

then $\varepsilon_1^2 \geq \varepsilon_2^2$.

This theorem is proved in Appendix III. We are now ready to compare the results of all transforms in terms of the visual, RMS, and correlated error criteria.

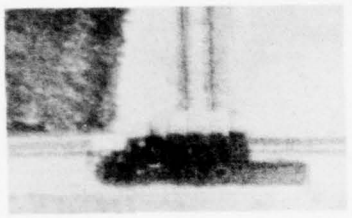
3. Comparison of Transforms

The truck scene was selected as representative of typical Remotely Piloted Vehicle image. The man-made structure as well as the background requires resolution performance to reproduce the gray level changes and edges. Other scenes available for this examination were considered spatially too flat. This scene is a 512 x 512, 6 bit digital image. The image was segmented into 1024, 16 x 16 blocks, or one dimensional vectors of length 256. The following transform processes were compared:

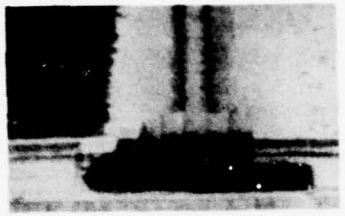
1. Hadamard
2. Fast Fourier
3. Slant
4. DLB (16 basis set)
5. DLB (8·2·8·2) basis set
6. DCT
7. Fast K-L.

a. Visual Criteria

Figures 4.1-4.9 depict the reconstructed images and their corresponding error images for compression ratios of 6:1, 8:1, and 12:1. In terms of average bits per picture element these are 1 bit/pel, .75 bit/pel and .5 bits/pel. There is, of course, a general degradation from 1 bit/pel to .5 bits/pel. This is most noticeable on the fender of the truck and by the loss of the exhaust pipe near the cab. The visual errors are more easily discerned for any one compression by the error images. The increase in detail of the bank, contrast on the railroad tracks and bridge are indicative of poorer performance. It appears, therefore, that all images are acceptable at the 1 bit/pel compression but only the DLB (8·2·8·2), Discrete Cosine, and Fast K-L Transforms are still adequately performing at the required .5 bits/pel. Table 4.1 classifies

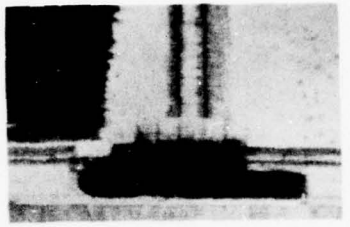


DLB(16)

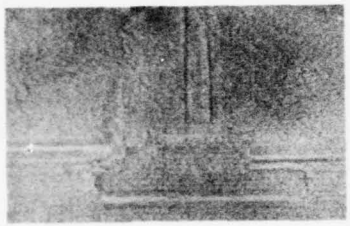


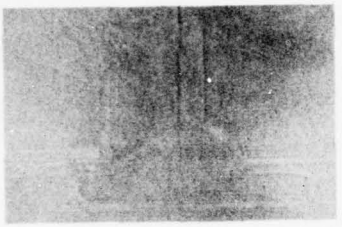
Reconstructed

DLB(8.2.8.2)

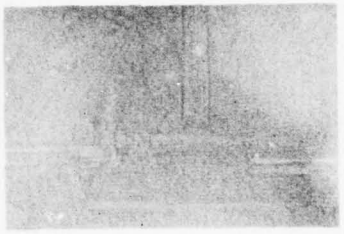


DCT





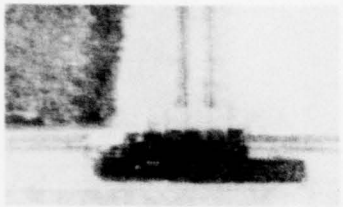
Error Image



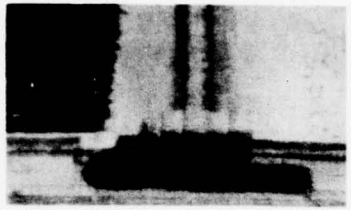
1 bit/pel

Comparison of the DLB(16), DLB(8.2.8.2), and the Discrete Cosine Transforms

Figure 4.1

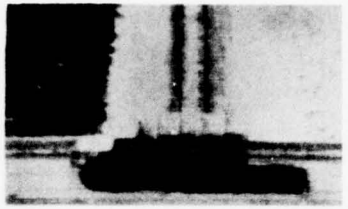


DLB(16)

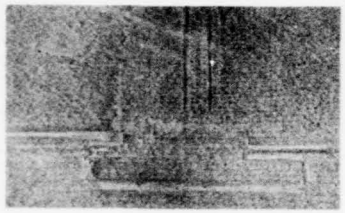


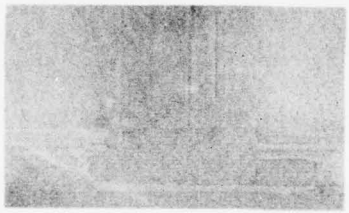
Reconstructed

DLB(8.2.8.2)

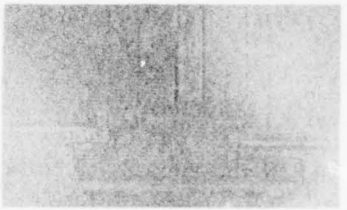


DCT





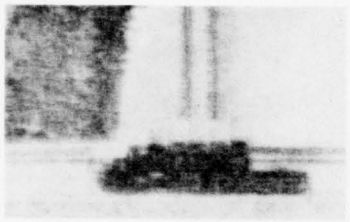
Error Image



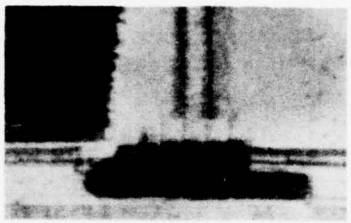
.75 bit/pel

Comparison of the DLB(16), DLB(8.2.8.2), and the Discrete Cosine Transforms

Figure 4.2

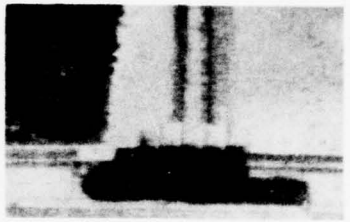


DLB (16)

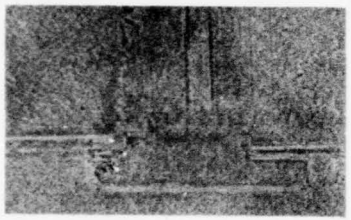


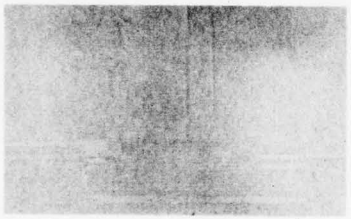
Reconstructed

DLB(8.2.8.2)



DCT



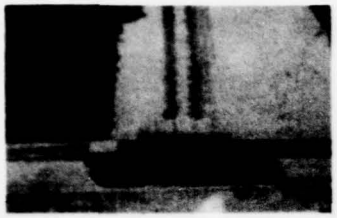


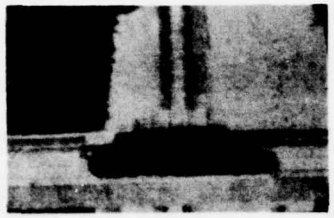
Error Image

.5 bit/pel

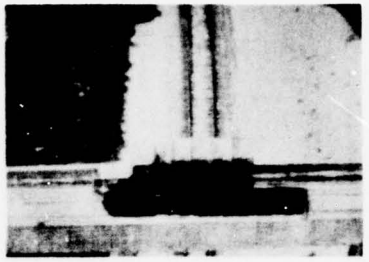
Comparison of the DLB(16), DLB(8.2.8.2), and the Discrete Cosine Transforms

Figure 4.3





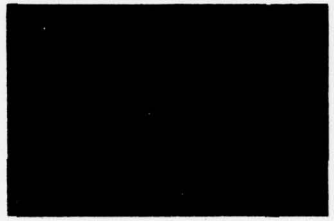
Reconstructed



Slant

Hadamard

Fourier

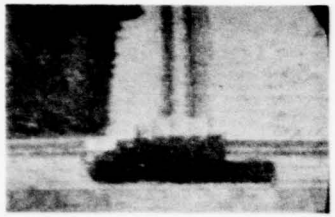


Error Image

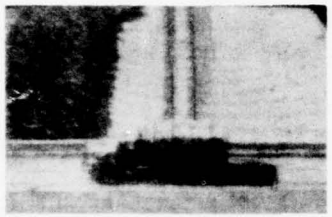
1 bit/pel

Comparison of Slant, Hadamard, and Fourier Transforms

Figure 4.4

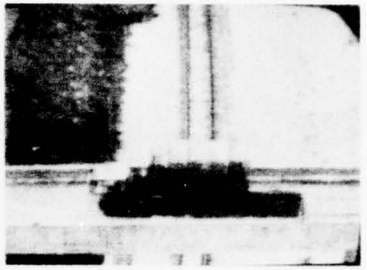


Slant

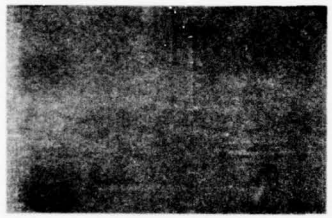


Reconstructed

Hadamard



Fourier

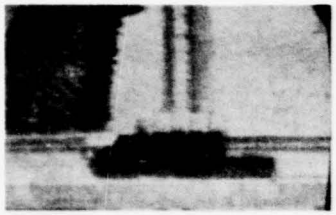


Error Image

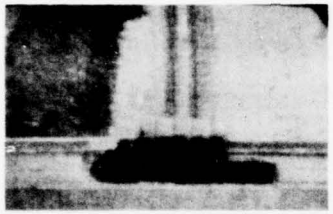
.75 bit/pel

Comparison of Slant,
Hadamard, and
Fourier Transforms

Figure 4.5

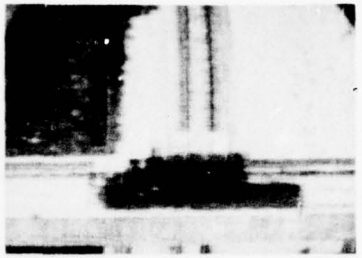


Slant



Reconstructed

Hadamard

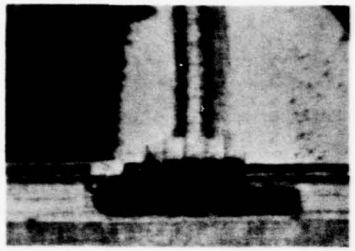


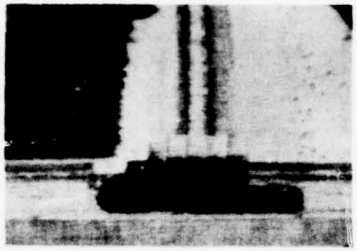
Fourier

Error Image

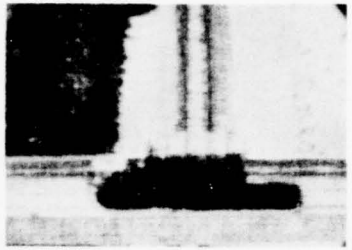
.5 bit/pel
Comparison of Slant,
Hadamard, and
Fourier Transforms

Figure 4.6





Reconstructed

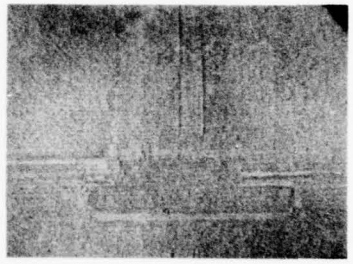


1 bit/pel

.75 bit/pel

.5 bit/pel

Error Image



Fast KL

Figures 4.7, 4.8, and 4.9

TABLE 4.1

Transform	Visual Comment	Tabulated Errors vs. Compression Ratio 1.0 bit/pel RMS	Lag Dist.	Correlation	Correlated Error
Fourier	Poor	4.0078	1	.10468879	.4195717326
		4.0078	2	.10909377	.4372260114
		4.0078	5	.10897077	.436733052
		4.0078	10	.10781116	.432085567
Hadamard	Fair	3.8528	1	.10758324	.4144967071
		3.8528	2	.11060267	.426129967
		3.8528	5	.11497447	.442973638
		3.8528	10	.1078733	.4156142502
Slant	Acceptable	3.7892	1	.10662527	.4040244731
		3.7892	2	.11079641	.4198290
		3.7892	5	.11075597	.4196765215
		3.7892	10	.10843463	.4108805
DLB (16x16)	Acceptable	3.9547	1	.1082334	.4280306
		3.9547	2	.1111686	.4396384
		3.9547	5	.11195713	.4427567
		3.9547	10	.1093633	.432499
DLB (8·2·8·2)	Good	3.7667	1	.10636617	.4006491
		3.7667	2	.10964318	.4129926
		3.7667	5	.10953416	.4125820
		3.7667	10	.1069852	.4029811
DCT	Good	3.6503	1	.10554689	.3852774
		3.6503	2	.11048985	.4033209
		3.6503	5	.10939932	.3993402
		3.6503	10	.10767923	.3930613
Fast K-L	Excellent to Good	3.6278	1	.10535227	.38219696
		3.6278	2	.10979803	.39832529
		3.6278	5	.10927618	.39643212
		3.6278	10	.10743651	.38975817

TABLE 4.1 (Continued)

Transform	Visual Comment	.75 bit/pel RMS	Lag Dist.	Tabulated Errors vs. Compression Ratio Correlation	Correlated Error
Fourier	Poor	4.3781	1	.10641426	.4658922717
		4.3781	2	.11174843	.4892458014
		4.3781	5	.11301188	.4947773118
		4.3781	10	.11126707	.4871383592
Hadamard	Poor	4.2293	1	.10843291	.4585953063
		4.2293	2	.11205909	.4739315093
		4.2293	5	.11222098	.4746161907
		4.2293	10	.10958335	.4634608622
Slant	Acceptable to Fair	4.1327	1	.10767135	.4449733881
		4.1327	2	.11225209	.4639042123
		4.1327	5	.11230641	.4641287006
		4.1327	10	.11046972	.4565382118
DLB (16x16)	Acceptable to Fair	4.2816	1	.10952806	.468955
		4.2816	2	.1135467	.4861615
		4.2816	5	.1141603	.4887887
		4.2816	10	.11178365	.4786126
DLB (8·2·8·2)	Good	4.0683	1	.10698853	.4352613
		4.0683	2	.11135341	.453019
		4.0683	5	.1119608	.4554901
		4.0683	10	.10978346	.4466318
DCT	Good	3.9727	1	.10626637	.4221641
		3.9727	2	.11134476	.442339
		3.9727	5	.11103255	.4410988
		3.9727	10	.10885344	.4324419
Fast K-L	Good	3.9567	1	.10573179	.4183489
		3.9567	2	.11113668	.43973450
		3.9567	5	.11053920	.43737045
		3.9567	10	.1089543	.4310994788

TABLE 4.1 (Continued)

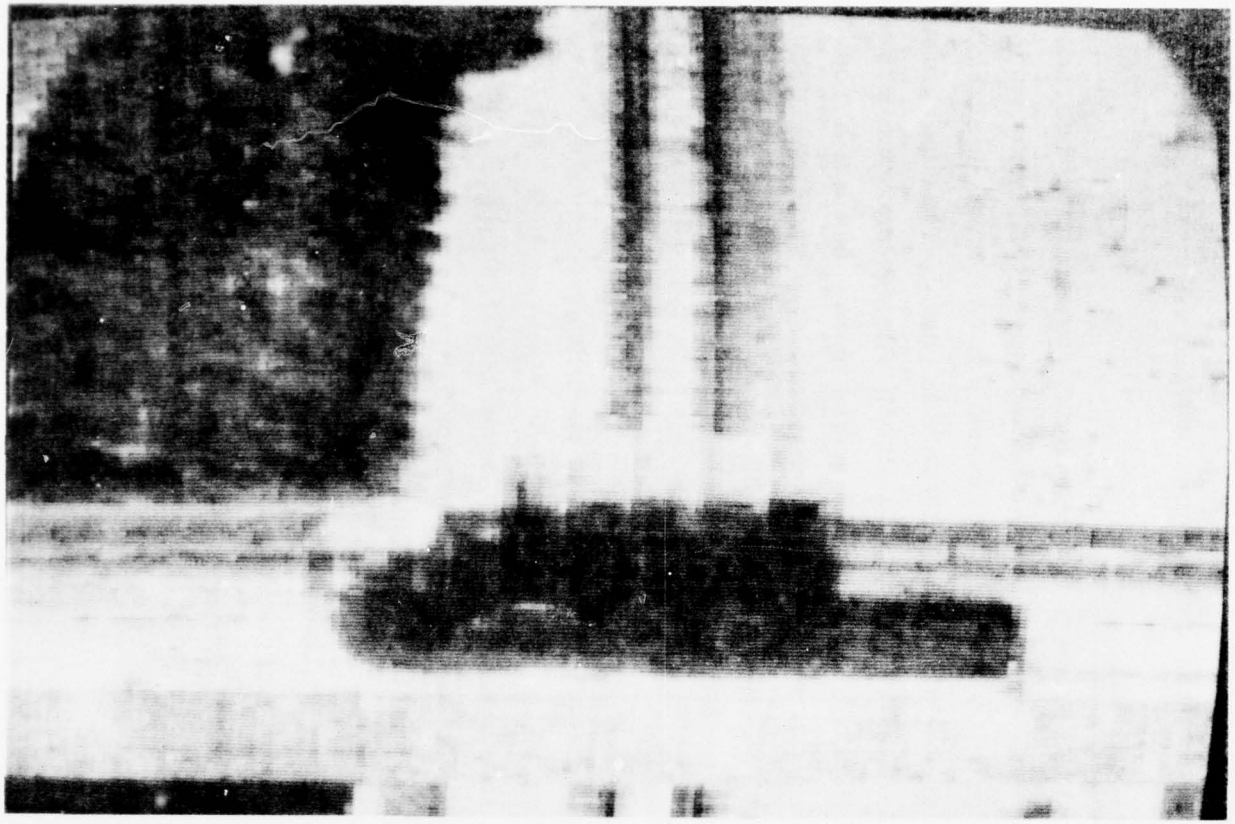
Transform	Visual Comment	.5 bit/pel RMS	Lag Dist.	Tabulated Errors vs. Compression Ratio Correlation	Correlated Error
Fourier	Poor	4.9412	1	.11130523	.5499814025
		4.9412	2	.11556605	.5710349663
		4.9412	5	.1179737	.5829316464
		4.9412	10	.11637595	.5750368441
Hadamard	Poor	4.7767	1	.11049775	.5278146024
		4.7767	2	.11396599	.5443813444
		4.7767	5	.11497440	.5491982165
		4.7767	10	.11288615	.5392232727
Slant	Acceptable to Poor	4.5942	1	.10924148	.5018772074
		4.5942	2	.11414770	.5244173633
		4.5942	5	.1156673	.5313987097
		4.5942	10	.11365033	.5221323461
DLB (16x16)	Acceptable to Poor	4.7318	1	.11010933	.5210151
		4.7318	2	.1144489	.4515493
		4.7318	5	.11584558	.5481577
		4.7318	10	.11358686	.53747
DLB (8·2·8·2)	Acceptable to Fair	4.5241	1	.10965364	.4960838
		4.5241	2	.1143836	.5174828
		4.5241	5	.1154832	.5224575
		4.5241	10	.1131099	.5117204
DCT	Good	4.43664	1	.10691045	.4743229
		4.43664	2	.11386058	.505158
		4.43664	5	.11446434	.5078368
		4.43664	10	.11266079	.4998349
Fast K-L	Good	4.4292	1	.10692673	.4735998
		4.4292	2	.11347968	.5026241987
		4.4292	5	.11412505	.5054826
		4.4292	10	.11246462	.498128294

the performance based on the compression ratio and quality. The subjective judgment can be aided by enlargement of these images. The .5 bit/pel images were blown up to see the effect of blocking in the image (Figures 4.10-4.16). This indicates that the DLB (8·2·8·2), Discrete Cosine, or Fast K-L is adequate for the higher compression ratios.

b. RMS Error Criteria

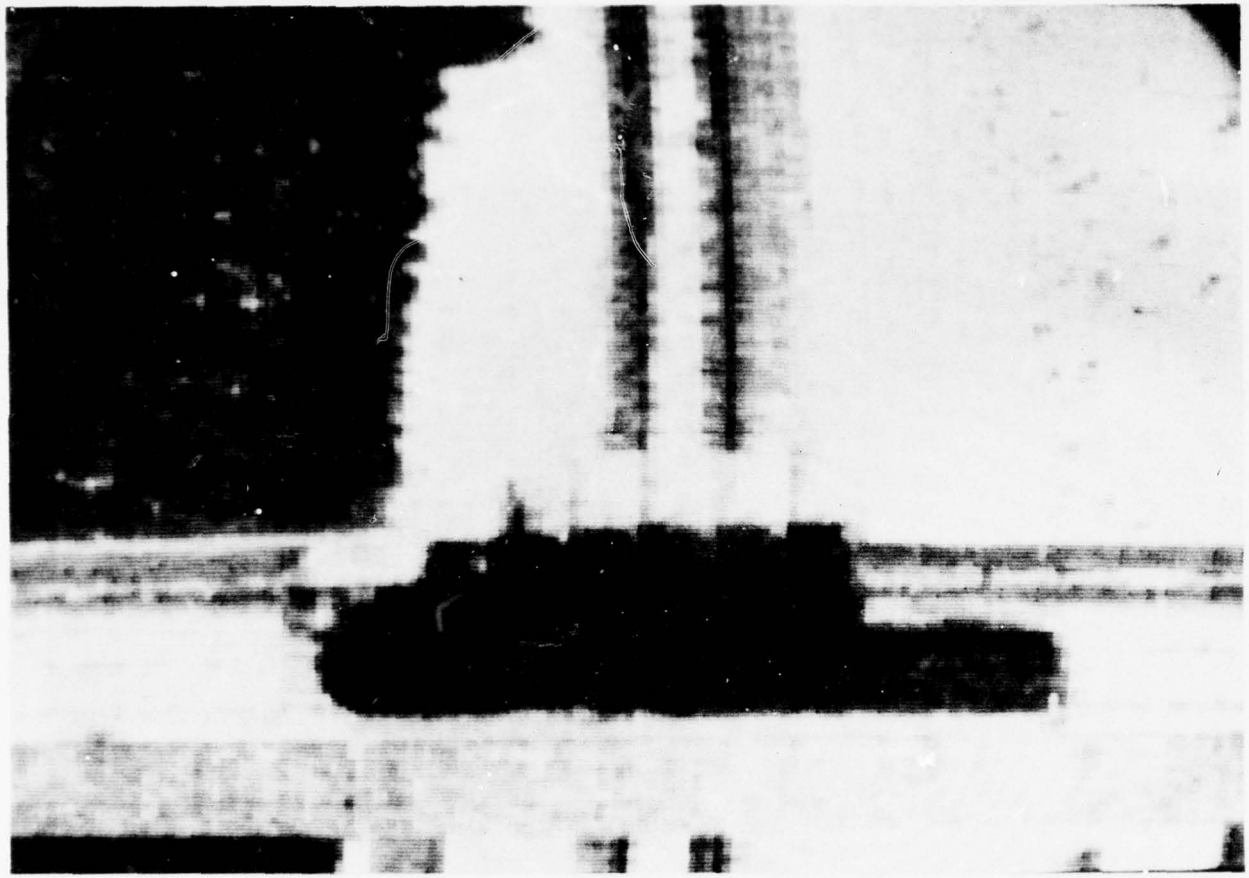
The second criteria we wish to investigate is the RMS error criteria. Figure 4.17 illustrates the error as a function of compression. In terms of the most optimum, the Fast Karhunen Loeve outperforms the rest. This performance is followed closely by the Discrete Cosine and the Discrete Linear Basis with the 8·2·8·2 basis set. Although the Fast K-L transform more closely approaches the Karhunen Loeve the basis set is highly dependent on the image and is rather lengthy to compute for the number of images to be used in the interframe process.

The optimality of the Fast K-L transform with just two layers infers that the other transforms would also have minimum error with the use of just two layers. This appears to be true for the Discrete Cosine but was not evident for the DLB. In the process of generating the DLB basis set, two variables must be selected (i.e., r,s). These two variables may take on an infinite number of positive and negative integer values. There is no guarantee, therefore, that the optimum has been selected. It is further true that the bit allocation assumes uncorrelated coefficients as input and since this particular set is more correlated than the other transforms tested, a poorer bit assignment may result.



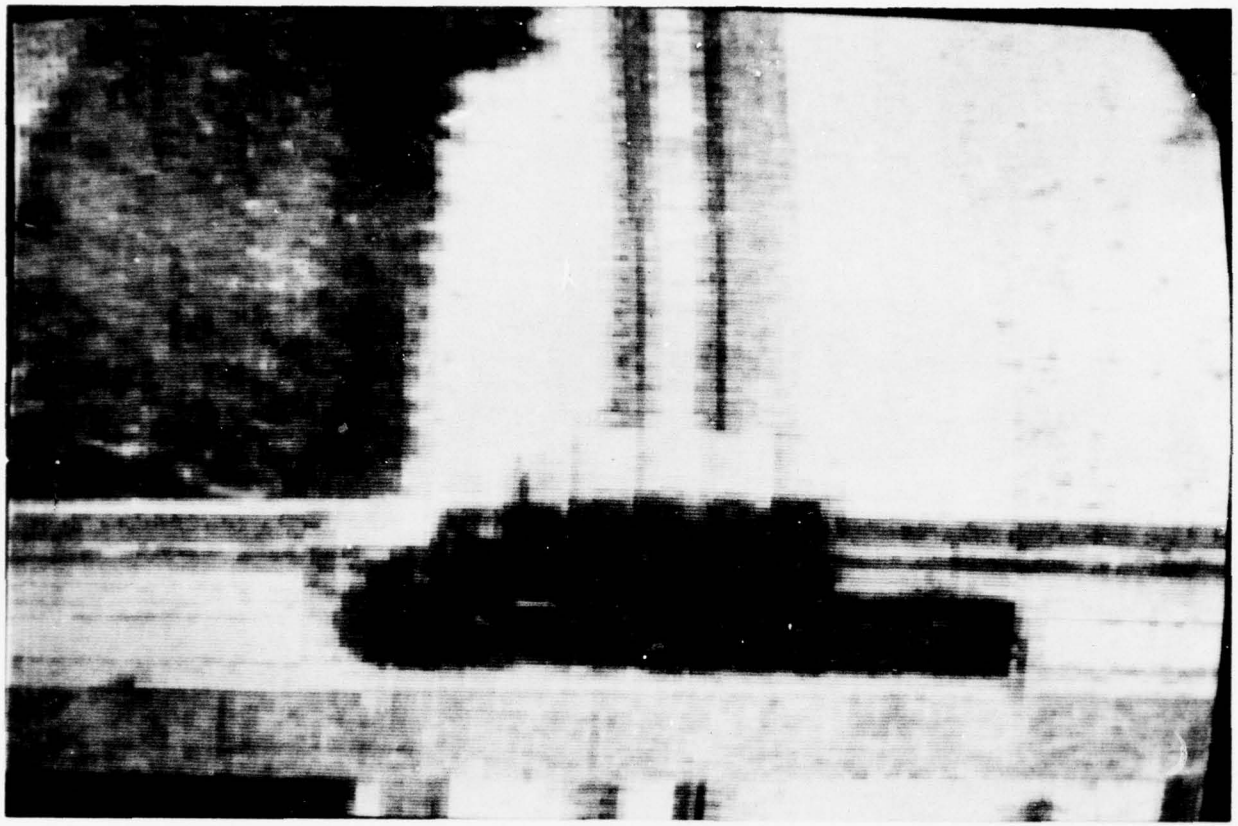
Enlargement of Slant
at .5 bit/pel

Figure 4.10



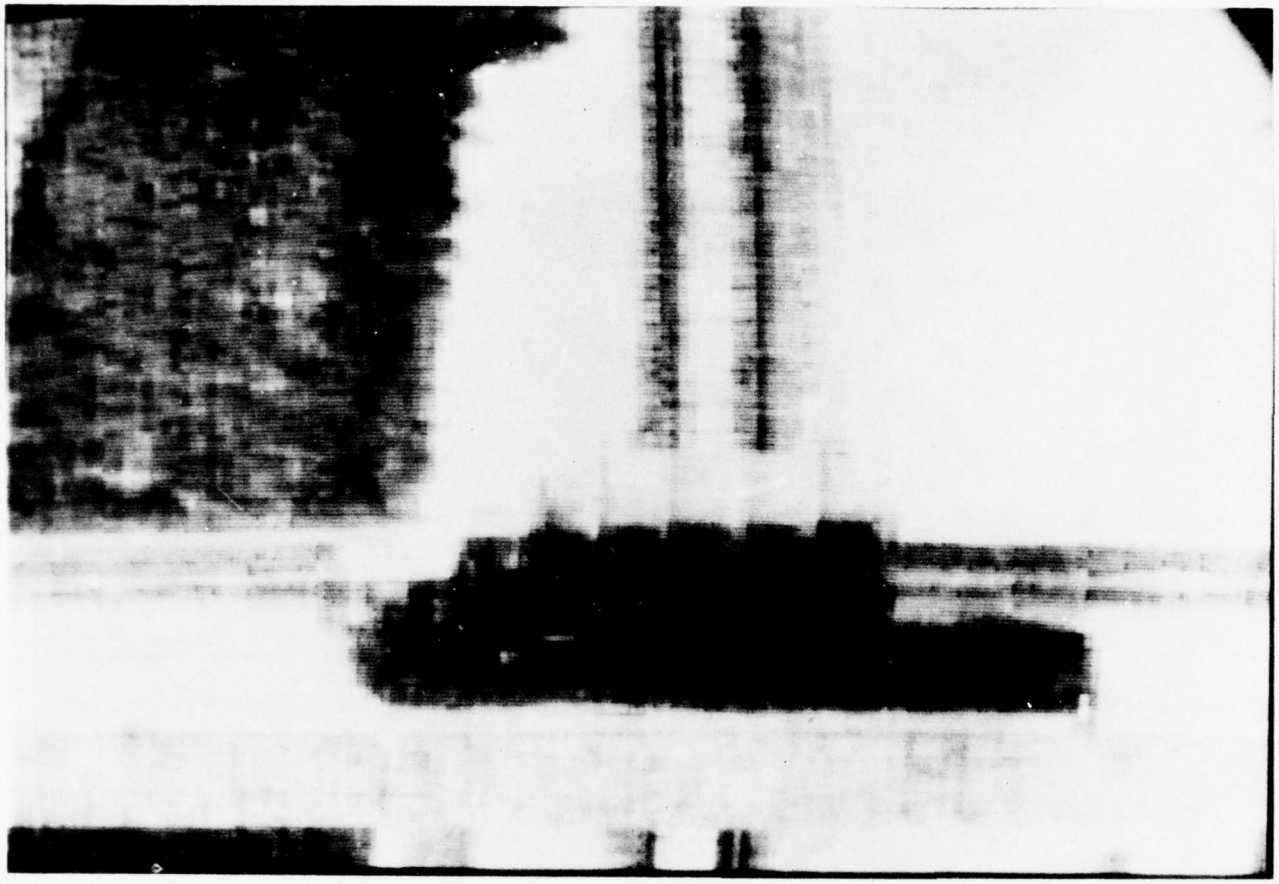
Enlargement of the Discrete Cosine
at .5 bit/pel

Figure 4.11



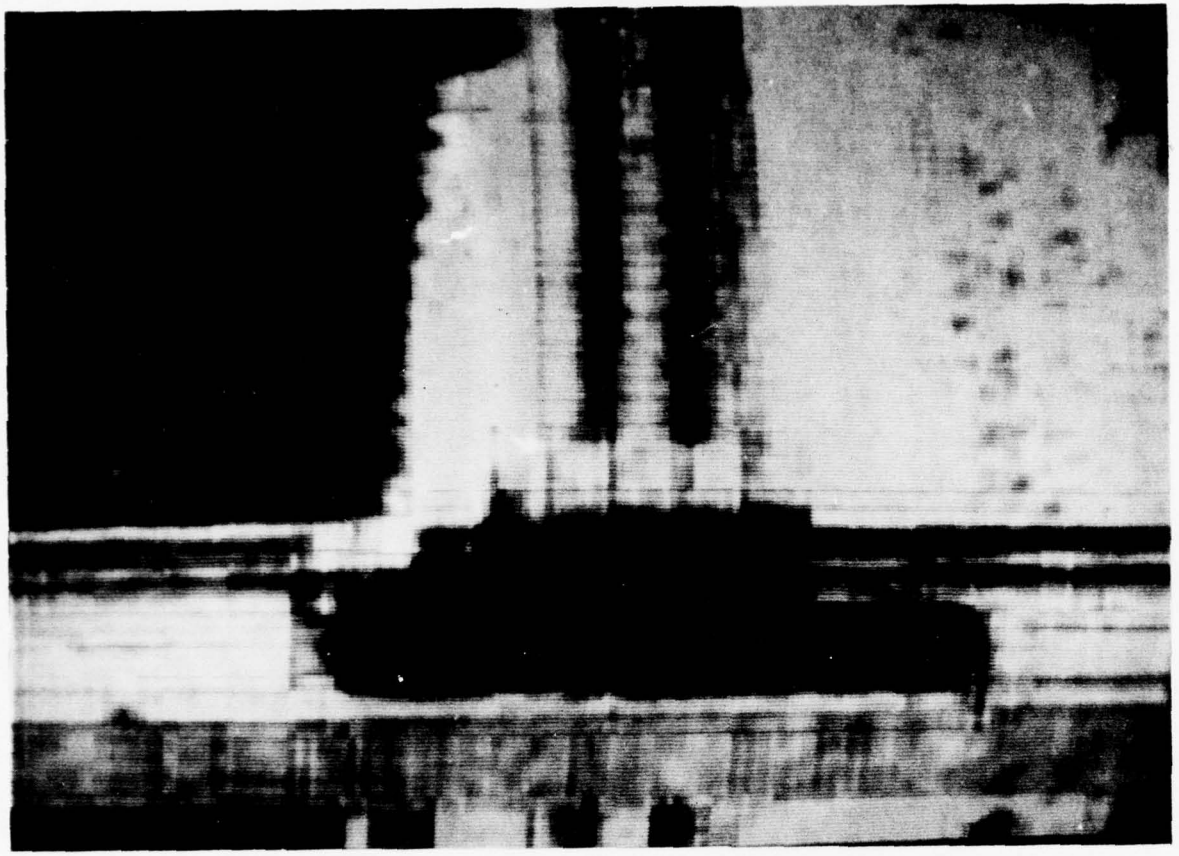
Enlargement of Hadamard
at .5 bit/pel

Figure 4.12



Enlargement of DLB(16)
at .5 bit/pel

Figure 4.13

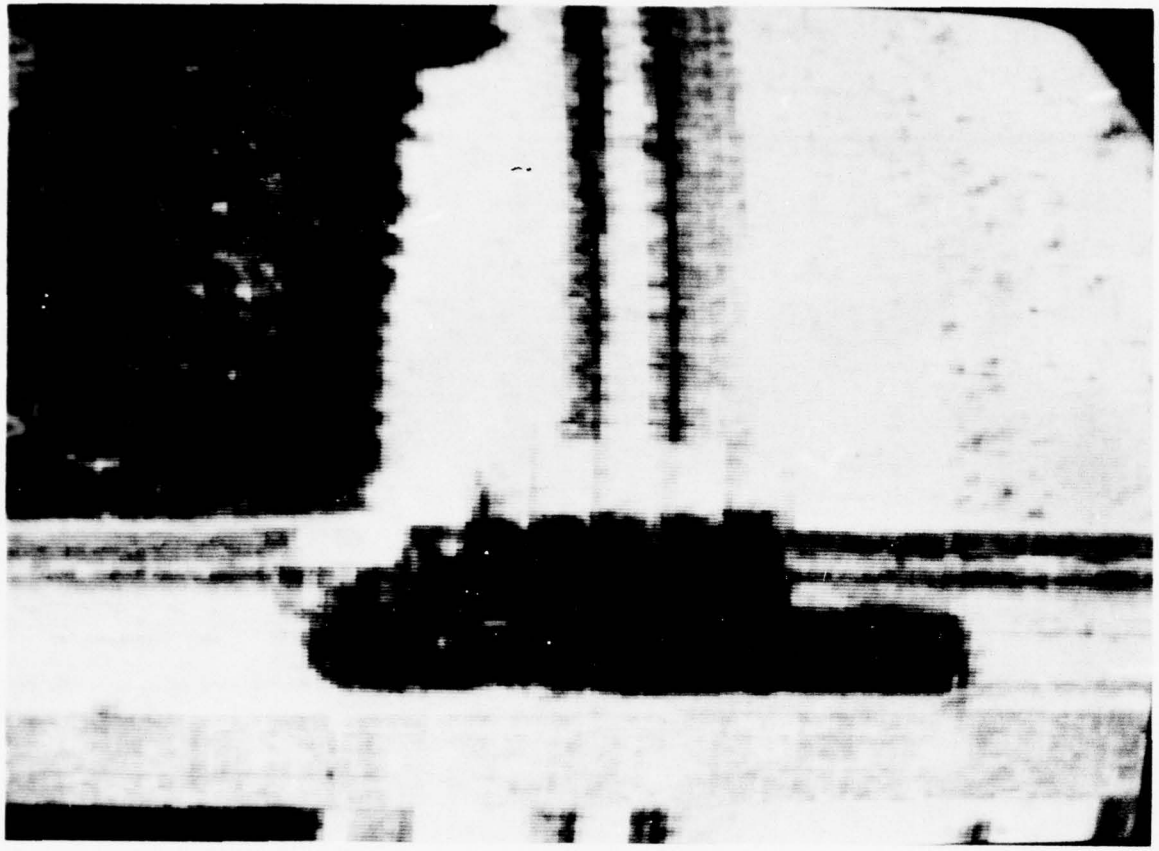


Enlargement of Fourier
at .5 bit/pel

Figure 4.14

Enlargement of DLB(8.2.8.2)
at .5 bit/pel

Figure 4.15



Enlargement of Fast KL
at .5 bit/pel

Figure 4.16

RMS Error as a Function of Compression

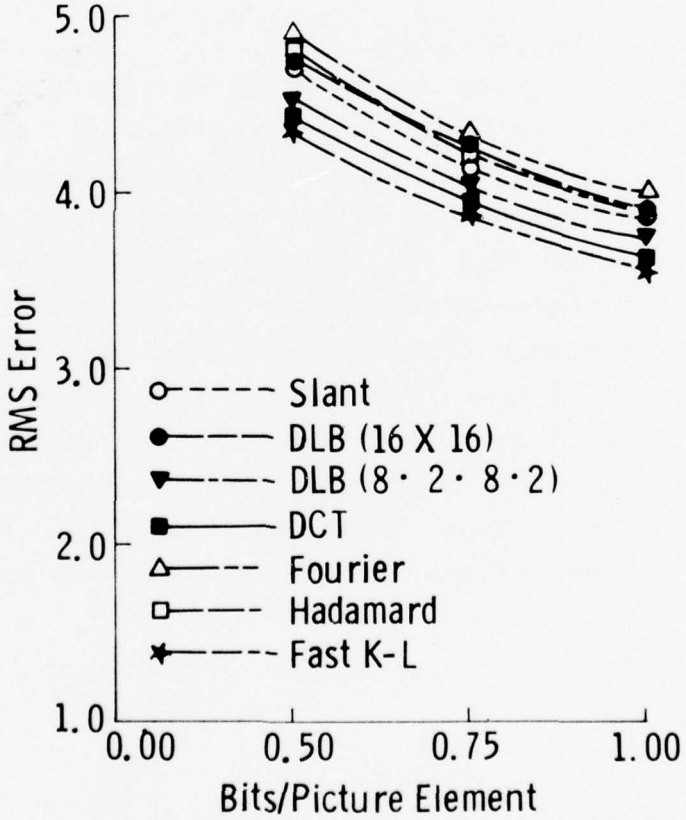


Figure 4.17

c. Correlation

To determine the correlated error in Table 4.1, the correlation values were calculated for each transform at lag distances of 1, 2, 5, 10. The spatial correlation of errors indicates that all transforms have less correlation at distance one and the Fast K-L is the most efficient in terms of this criteria. The Discrete Cosine also has low correlation at all distances compared to the other transforms. This is illustrated in Figures 4.18-4.20. The correlation is plotted as a function of lag distance for the three compression ratios.

d. Basis Vector Comparisons

In addition to looking at the error criteria, it is informative to investigate the sequency properties of the basis vectors used. The discrete transforms generally exhibit zero crossing properties called sequence. It is suggested in the literature [2] that as the subimage size grows larger the Discrete Cosine basis set approaches a correlation matrix with a first order Markov assumption. This is, however, only true if the image approaches a Markov Process. To clarify this suggestion in the literature a correlation matrix of a first order Markov assumption was generated with a correlation coefficient equal to 0.95. This matrix is a class of toeplitz matrices of the form

$$\psi = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{M-1} \\ \rho & 1 & \rho & & \rho^{M-2} \\ \vdots & & & & \\ \rho^{M-1} & \rho^{M-2} & \cdots & & 1 \end{bmatrix} \quad 0 < \rho < 1$$

M is equal to the subimage size.

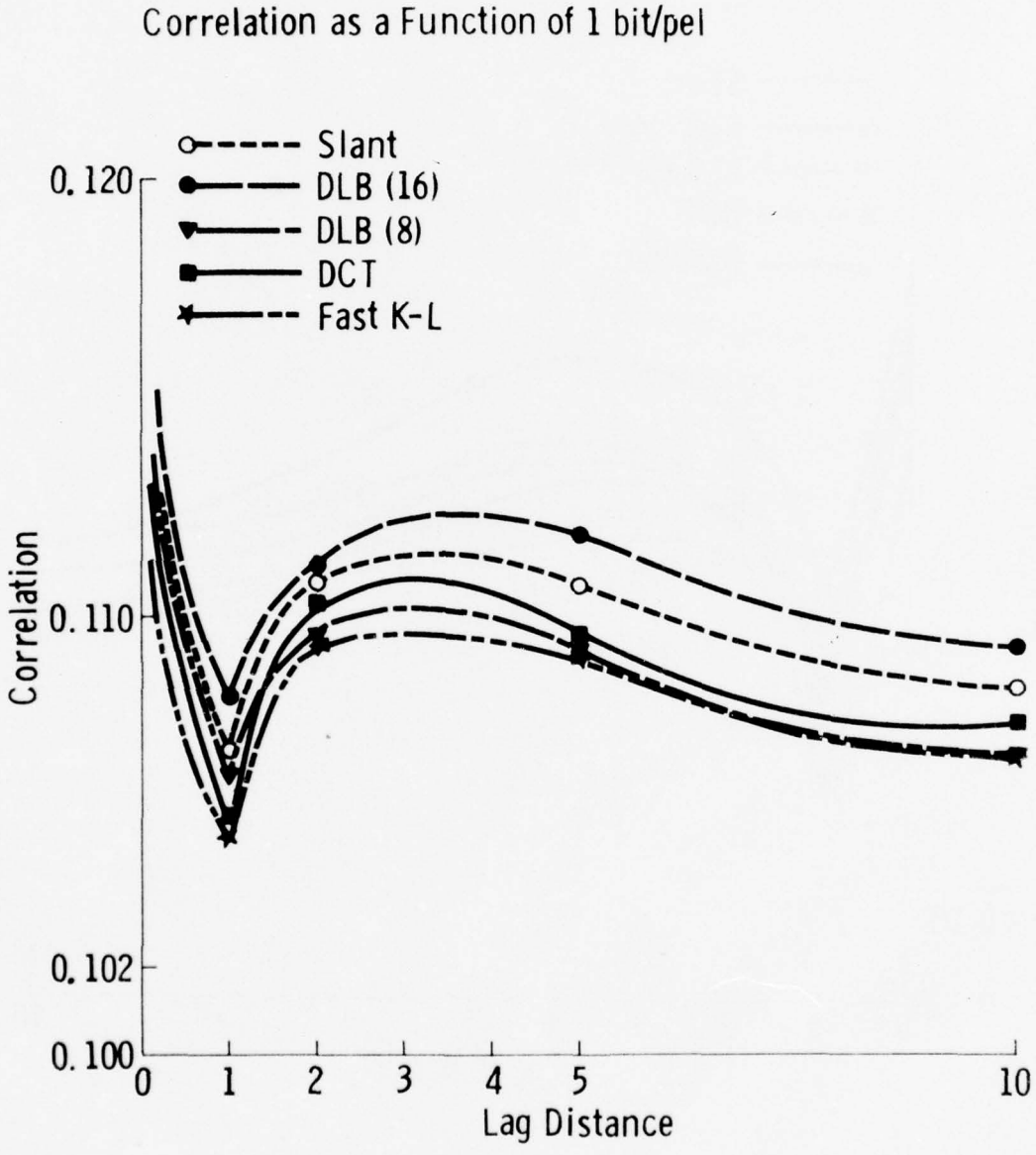

Correlation as a Function of 1 bit/pel
Slant
DLB (16)
DLB (8)
DCT
Fast K-L
0.120
0.110
0.102
0.100
Correlation
0
1
2
3
4
5
10
Lag Distance

Figure 4.18

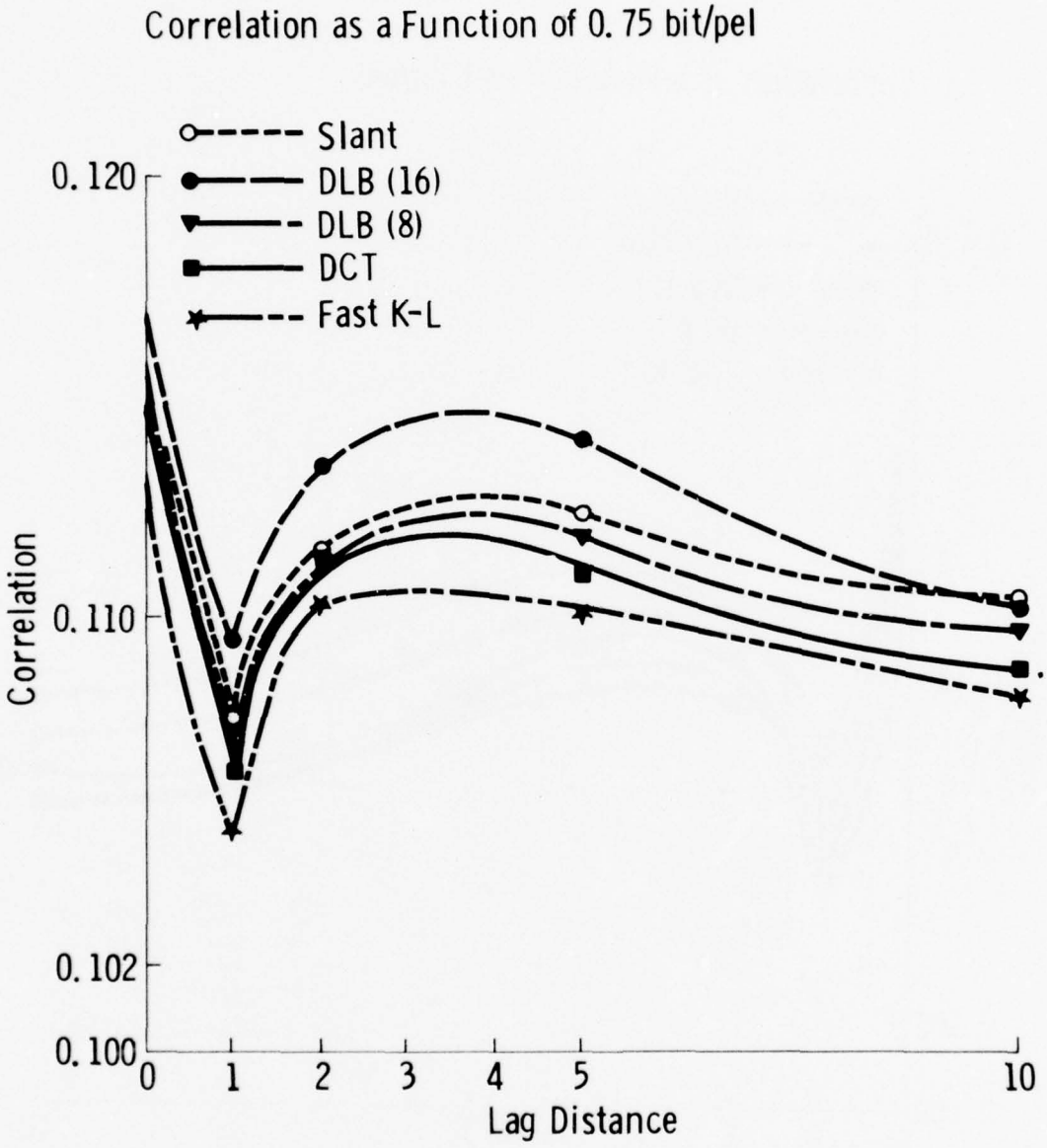

Correlation as a Function of 0.75 bit/pel
Slant
DLB (16)
DLB (8)
DCT
Fast K-L
0.120
0.110
0.102
0.100
Correlation
0
1
2
3
4
5
10
Lag Distance

Figure 4.19

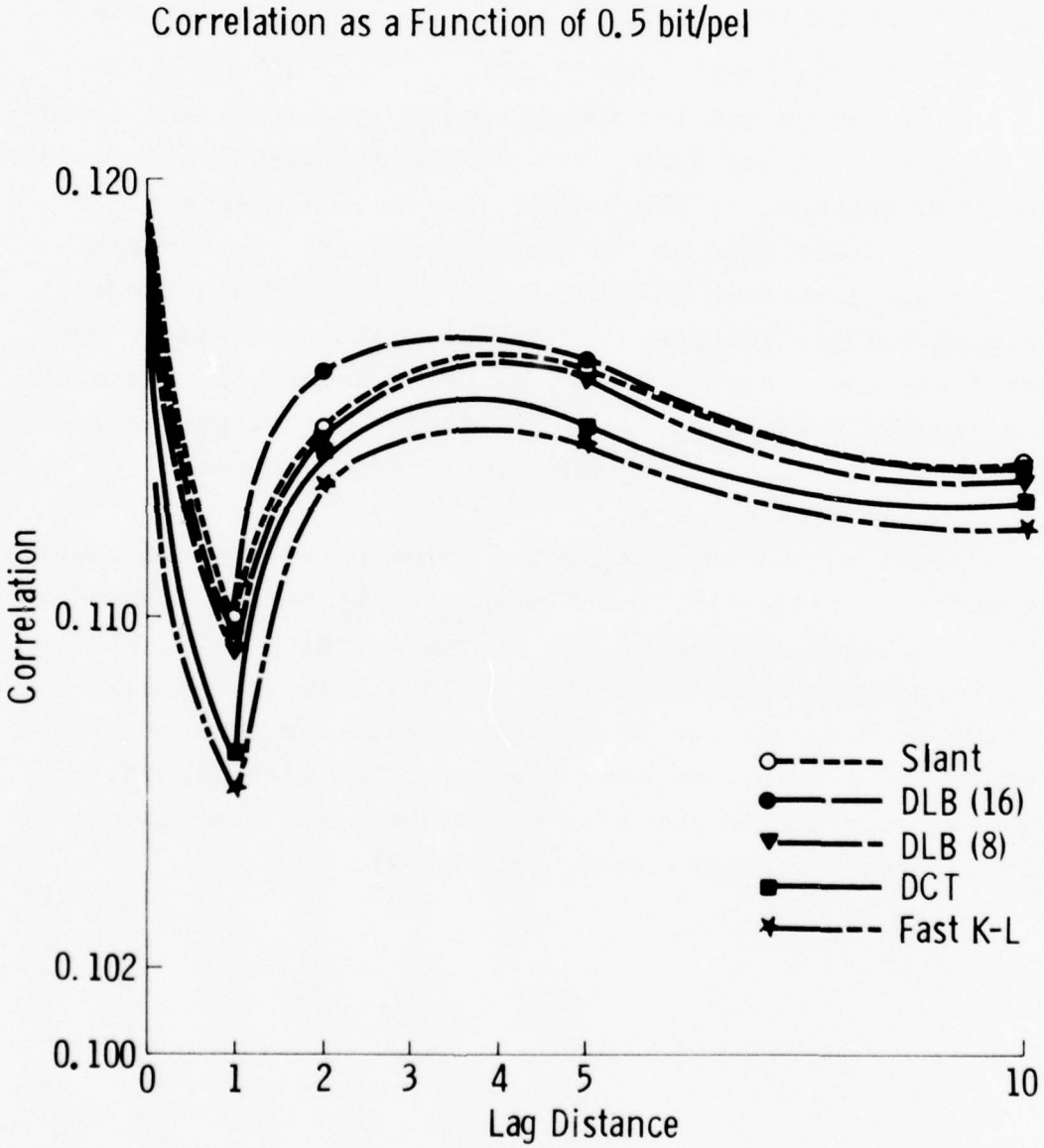

Correlation as a Function of 0.5 bit/pel
0.120
0.110
0.102
0.100
Correlation
0
1
2
3
4
5
10
Lag Distance
Slant
DLB (16)
DLB (8)
DCT
Fast K-L

Figure 4.20

The first order assumption gives the relation of gray level dependency of the resolution cells. If the subimage is considered as a vector and we know the conditional probability of random variables X_1, X_2,...,X_n, and for any n we have Pr. $(X_n|X_{n-1}, X_{n-2},\ldots,X_1)$ = Pr. $(X_n|X_{n-1})$.

This matrix and its normalized eigenvectors are given in Figures 4.21 and 4.22. The Discrete Cosine basis vectors are then compared to the Markov correlation matrix eigenvectors. Other than an obvious phase shift the vectors are almost identical in terms of sequency and magnitude (Figure 4.23). The other transforms which are easily compared are the Discrete Linear Basis, Hadamard and Slant. See Figures 4.24, 4.25, 4.26 and 4.27. Notice particularly the variation in sequency one, and fifteen compared to the Discrete Cosine.

Based on the results of the error criterion and results of sequence plots the "best" transform to be implemented in the interframe process is the Discrete Cosine. The Fast K-L transform was not utilized based solely on the time factor to generate the basis vector sets for each of the 12 images to be used. Further investigation with regard to classes of pictures for this transform must first be completed (see Recommendations, Chapter 7).

1.000000 0.950000 0.902500 0.857375 0.814506 0.773781 0.735092 0.698337
0.663420 0.630249 0.598737 0.568800 0.540360 0.513342 0.487675 0.463291
0.950000 1.000000 0.950000 0.902500 0.857375 0.814506 0.773781 0.735092
0.698337 0.663420 0.630249 0.598737 0.568800 0.540360 0.513342 0.487675
0.902500 0.950000 1.000000 0.950000 0.902500 0.857375 0.814506 0.773781
0.735092 0.698337 0.663420 0.630249 0.598737 0.568800 0.540360 0.513342
0.857375 0.902500 0.950000 1.000000 0.950000 0.902500 0.857375 0.814506
0.773781 0.735092 0.698337 0.663420 0.630249 0.598737 0.568800 0.540360
0.814506 0.857375 0.902500 0.950000 1.000000 0.950000 0.902500 0.857375
0.814506 0.773781 0.735092 0.698337 0.663420 0.630249 0.598737 0.568800
0.773781 0.814506 0.857375 0.902500 0.950000 1.000000 0.950000 0.902500
0.857375 0.814506 0.773781 0.735092 0.698337 0.663420 0.630249 0.598737
0.735092 0.773781 0.814506 0.857375 0.902500 0.950000 1.000000 0.950000
0.902500 0.857375 0.814506 0.773781 0.735092 0.698337 0.663420 0.630249
0.698337 0.735092 0.773781 0.814506 0.857375 0.902500 0.950000 1.000000
0.950000 0.902500 0.857375 0.814506 0.773781 0.735092 0.698337 0.663420
0.663420 0.698337 0.735092 0.773781 0.814506 0.857375 0.902500 0.950000
1.000000 0.950000 0.902500 0.857375 0.814506 0.773781 0.735092 0.698337
0.630249 0.663420 0.698337 0.735092 0.773781 0.814506 0.857375 0.902500
0.950000 1.000000 0.950000 0.902500 0.857375 0.814506 0.773781 0.735092
0.598737 0.630249 0.663420 0.698337 0.735092 0.773781 0.814506 0.857375
0.902500 0.950000 1.000000 0.950000 0.902500 0.857375 0.814506 0.773781
0.568800 0.598737 0.630249 0.663420 0.698337 0.735092 0.773781 0.814506
0.857375 0.902500 0.950000 1.000000 0.950000 0.902500 0.857375 0.814506
0.540360 0.568800 0.598737 0.630249 0.663420 0.698337 0.735092 0.773781
0.814506 0.857375 0.902500 0.950000 1.000000 0.950000 0.902500 0.857375
0.513342 0.540360 0.568800 0.598737 0.630249 0.663420 0.698337 0.735092
0.773781 0.814506 0.857375 0.902500 0.950000 1.000000 0.950000 0.902500
0.487675 0.513342 0.540360 0.568800 0.598737 0.630249 0.663420 0.698337
0.735092 0.773781 0.814506 0.857375 0.902500 0.950000 1.000000 0.950000
0.463291 0.487675 0.513342 0.540360 0.568800 0.598737 0.630249 0.663420
0.698337 0.735092 0.773781 0.814506 0.857375 0.902500 0.950000 1.000000

Figure 4.21. First Order Markov Correlation Matrix $\rho = .95$ (16x16) Each row equals two printed lines.

```
 0.035459-0.070570-0.104998 0.138410-0.170472-0.200875-0.229321-0.255507
  -0.279172 0.300036-0.317801-0.332085-0.342169 0.345817 0.331757-0.224355
-0.103247 0.197472 0.274467-0.327489 0.351852 0.345367 0.308511 0.244363
   0.158417-0.058043-0.048060-0.150554-0.239974 0.306306 0.331719-0.234313
 0.167079-0.294405-0.351722 0.325342-0.221485-0.064820 0.107357 0.253948
   0.339774-0.343948 0.264876 0.120804-0.054102 0.216483 0.315059-0.242954
-0.224511 0.346658 0.310806-0.133221-0.105121-0.295502-0.350950-0.245987
  -0.028436-0.202168 0.339749 0.320085 0.150569 0.091102 0.282610-0.250231
 0.273341-0.346303-0.165465-0.136690 0.338586 0.292155 0.031381-0.252377
  -0.350653 0.190999 0.109265 0.328391 0.302814-0.049244 0.235999-0.256102
-0.311700 0.293394-0.035470 0.326786-0.272057 0.070743 0.338544 0.247599
  -0.105706 0.346667-0.219423 0.140766 0.349454-0.181500 0.177561-0.260534
 0.338116-0.195961 0.224488-0.326068-0.035521-0.346578-0.165215 0.250796
   0.310215 0.071270-0.350900-0.130846 0.274266-0.283945 0.110225-0.263503
-0.351579 0.068787-0.338078 0.134959 0.311630 0.195882-0.273228-0.249203
   0.224379-0.292747-0.166998-0.324390 0.103440-0.339750 0.037366-0.264991
 0.351574 0.068827 0.338077 0.134951-0.311631 0.195886 0.273226-0.249203
  -0.224380-0.292747 0.166998-0.324390-0.103440-0.339750-0.037366-0.264991
-0.338101-0.195998-0.224486-0.326060 0.035521-0.346582 0.165216 0.250794
  -0.310215 0.071271 0.350900-0.130846-0.274266-0.283945-0.110225-0.263503
 0.311675 0.293427 0.035470 0.326777 0.272060 0.070747-0.338541 0.247602
   0.105707 0.346666 0.219423 0.140766-0.349454-0.181500-0.177561-0.260534
-0.273311-0.346330 0.165467-0.136683-0.338586 0.292152-0.031381-0.252376
   0.350654 0.190998-0.109265 0.328391-0.302814-0.049244-0.235999-0.256102
 0.224476 0.346677-0.310812-0.133223 0.105116-0.295503 0.350945-0.245990
   0.028434-0.202168-0.339749 0.320085-0.150569 0.091102-0.282610-0.250231
-0.167043-0.294416 0.351730 0.325340 0.221490-0.064818-0.107355 0.253947
  -0.339776-0.343947-0.264876 0.120803 0.054102 0.216483-0.315059-0.242954
 0.103216 0.197476-0.274474-0.327487-0.351854 0.345368-0.308507 0.244370
  -0.158415-0.058042 0.048060-0.150554 0.239974 0.306306-0.331719-0.234313
-0.035446-0.070571 0.105001 0.138409 0.170473-0.200877 0.229318-0.255511
   0.279172 0.300034 0.317801-0.332085 0.342169 0.345817-0.331757-0.224355
```

Figure 4.22. Eigenvectors of Correlation Matrix read in columns alternating rows, i.e., 1st vector contains 1st element of rows 1, 3, 5, 7 etc.

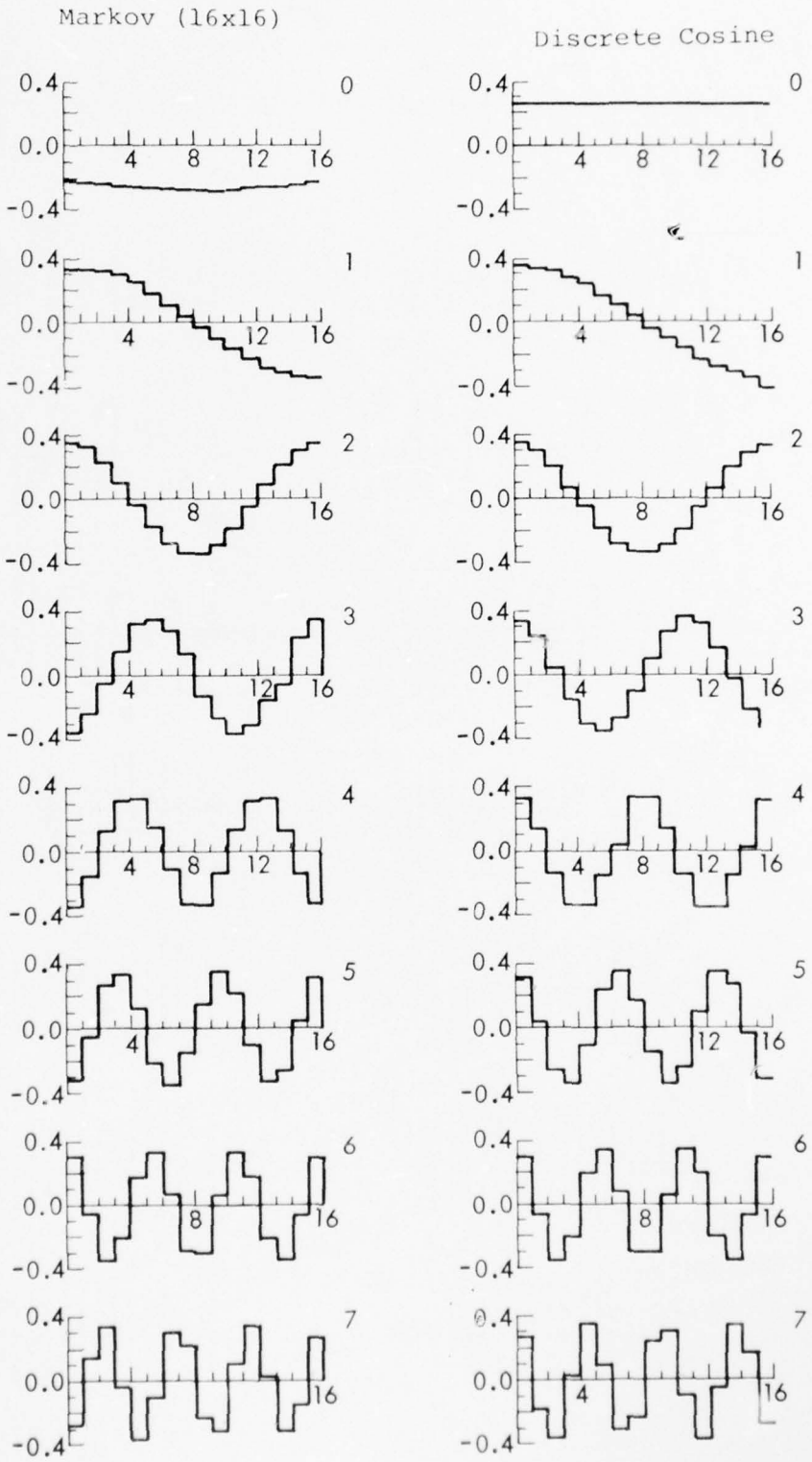


Figure 4.23a. Sequency Comparison

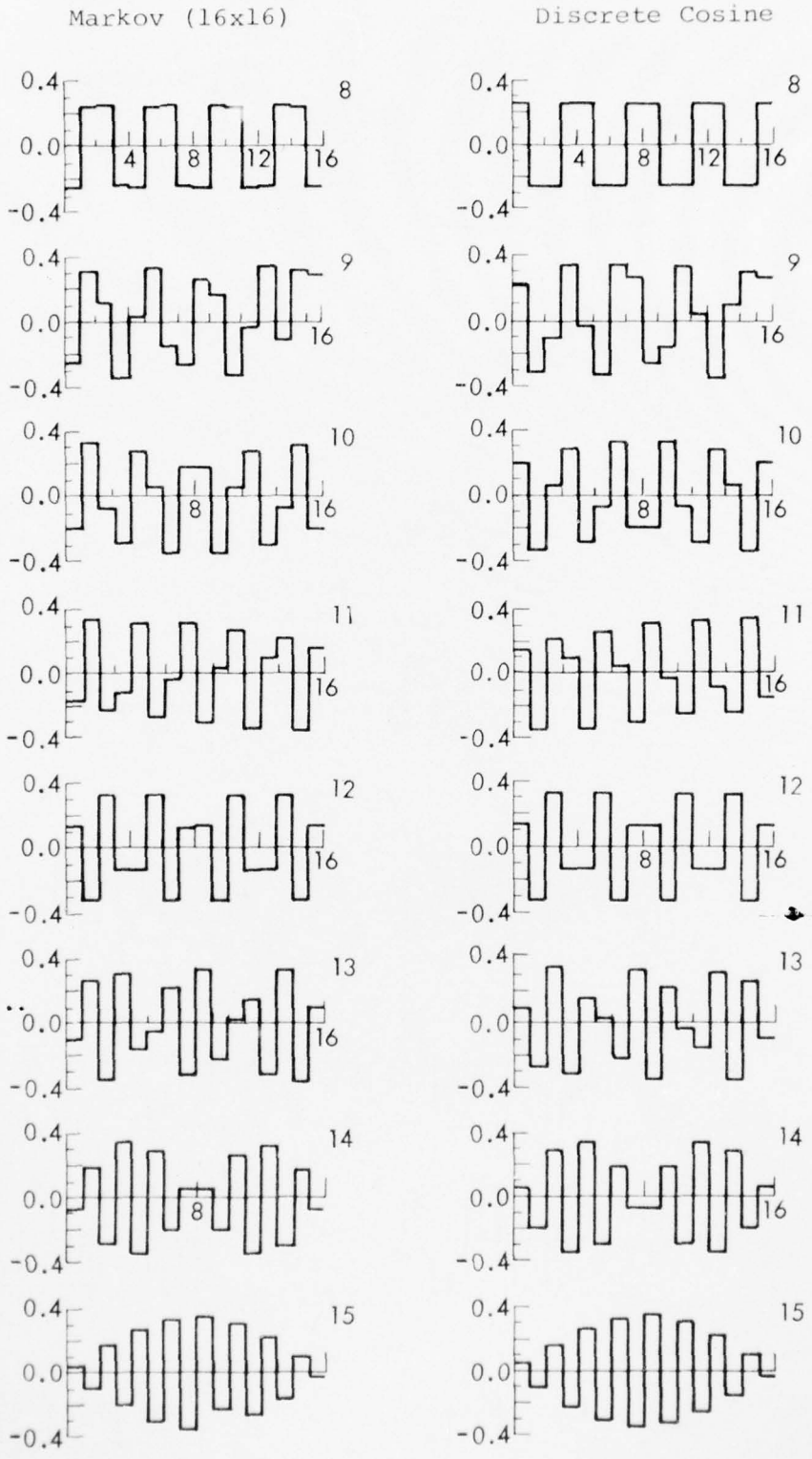


Figure 4.23b. Sequency Comparison

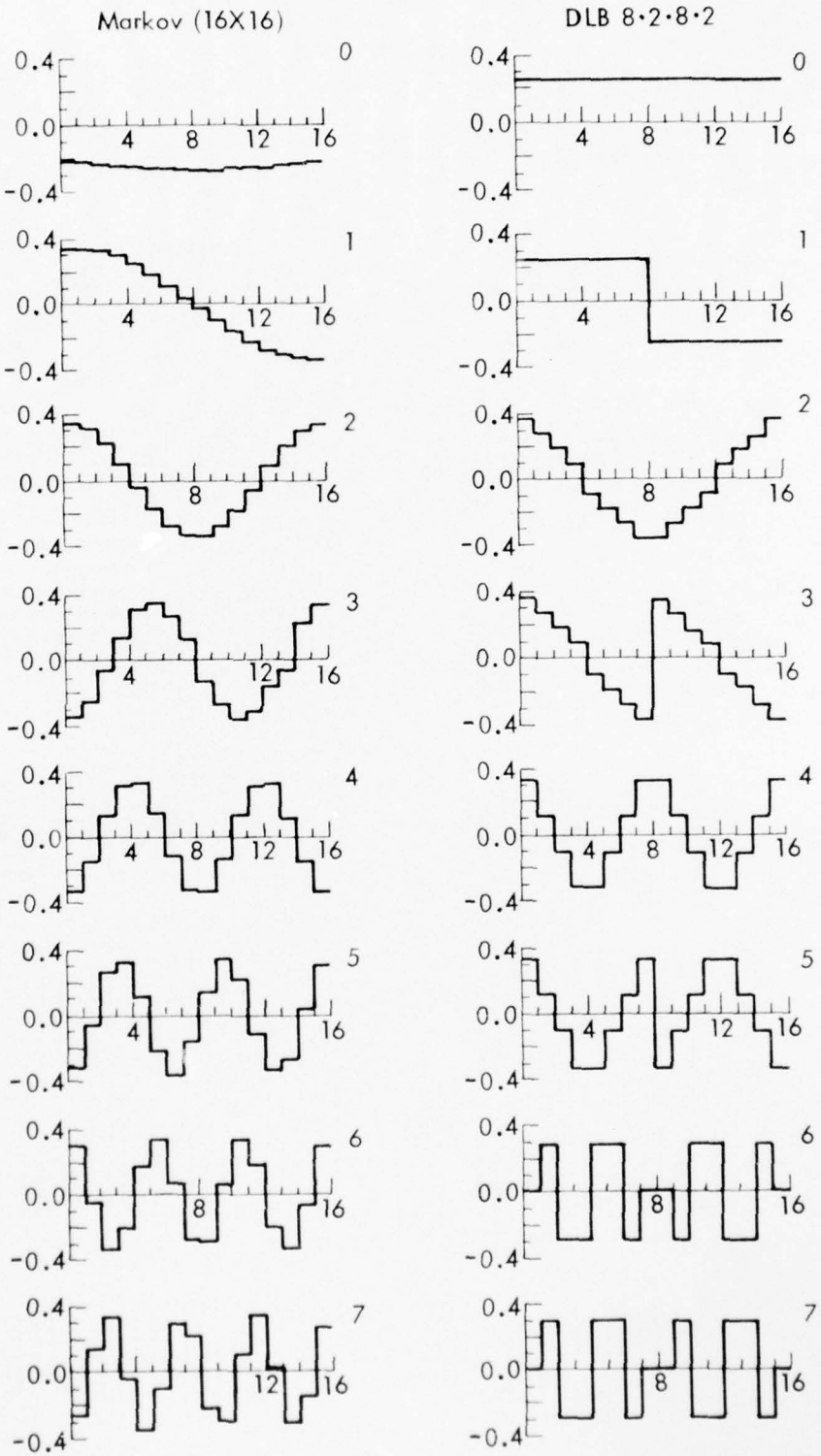


Figure 4.24a. Sequency Comparison

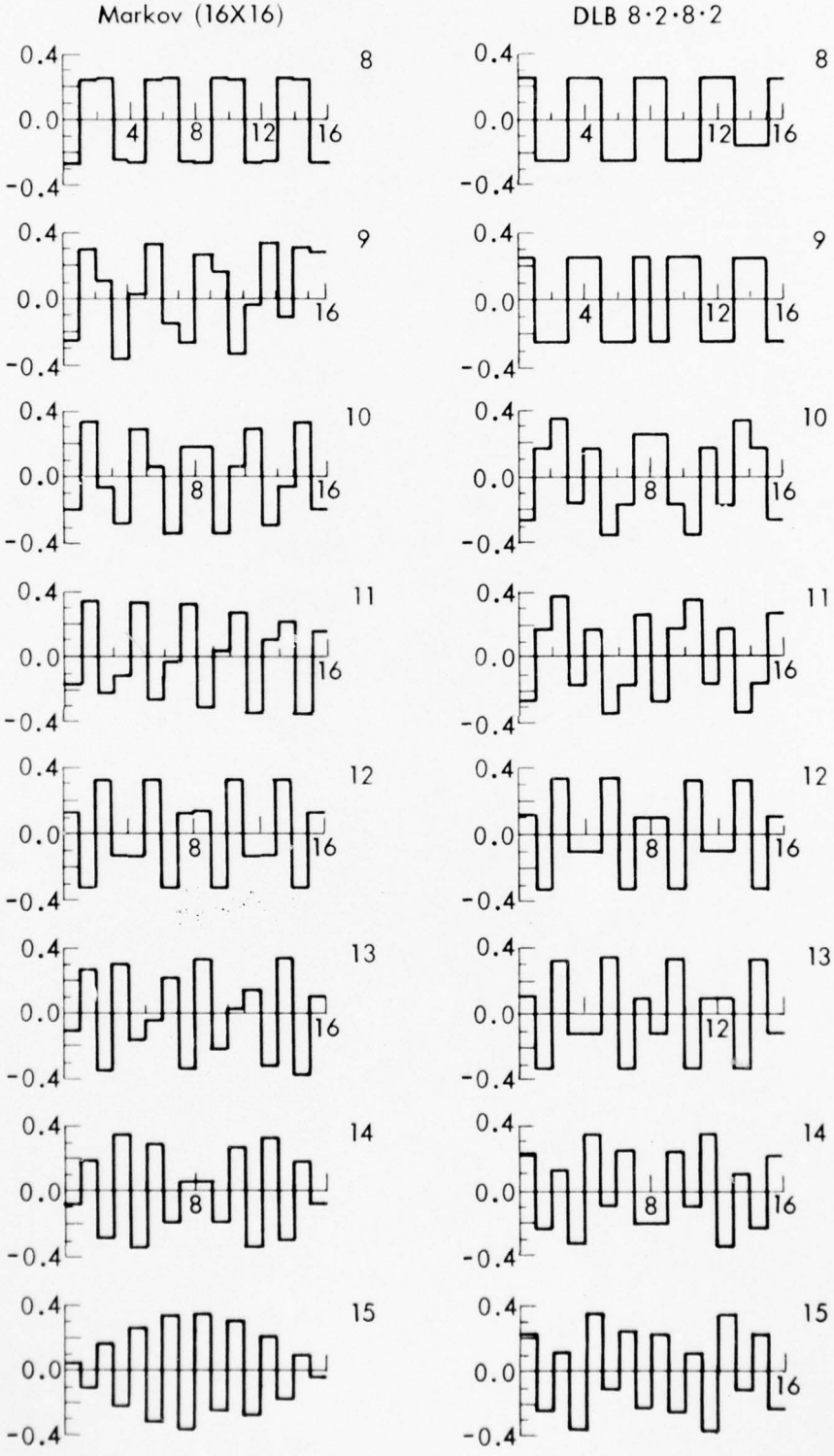


Figure 4.24b. Sequency Comparison

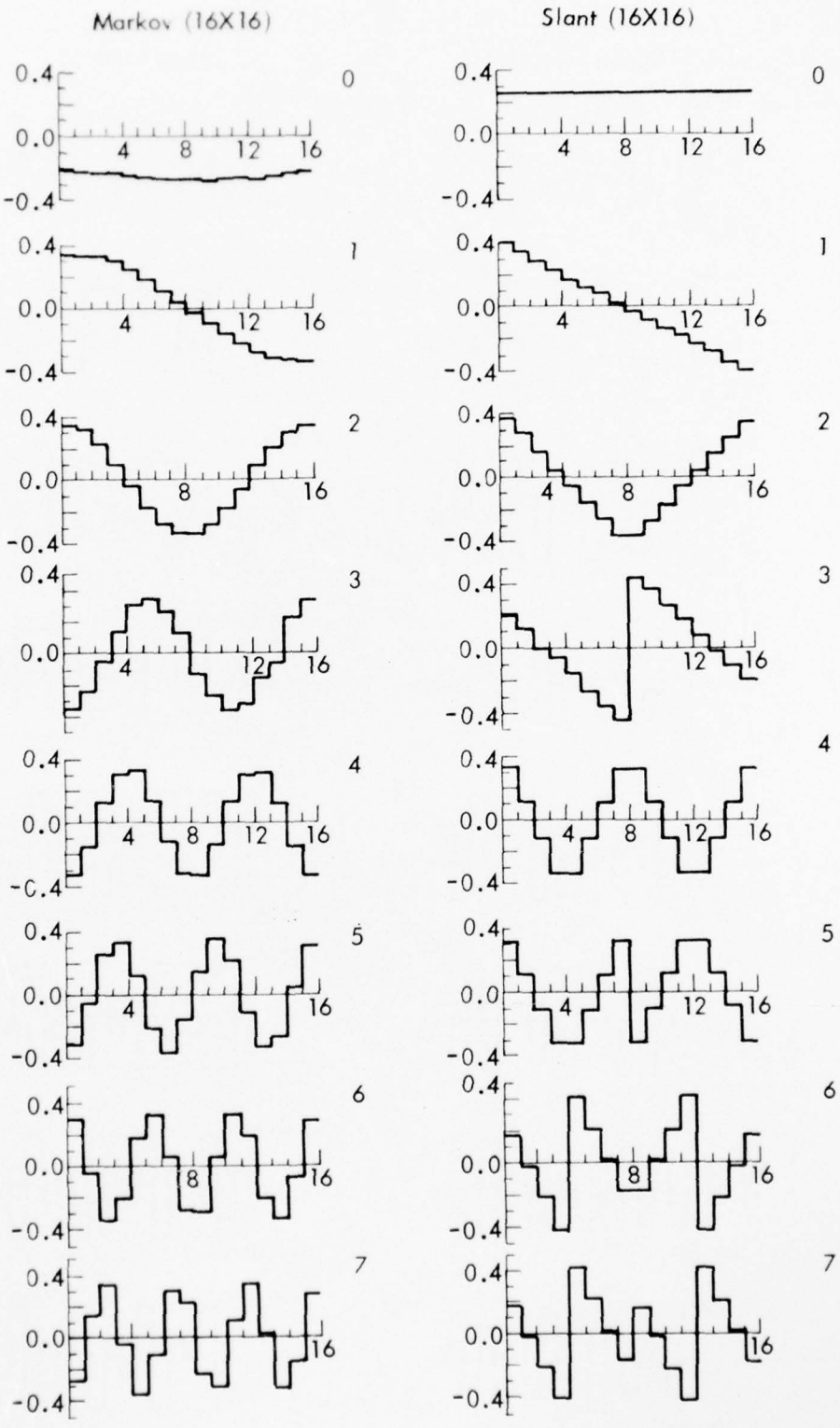


Figure 4.25a. Sequency Comparison

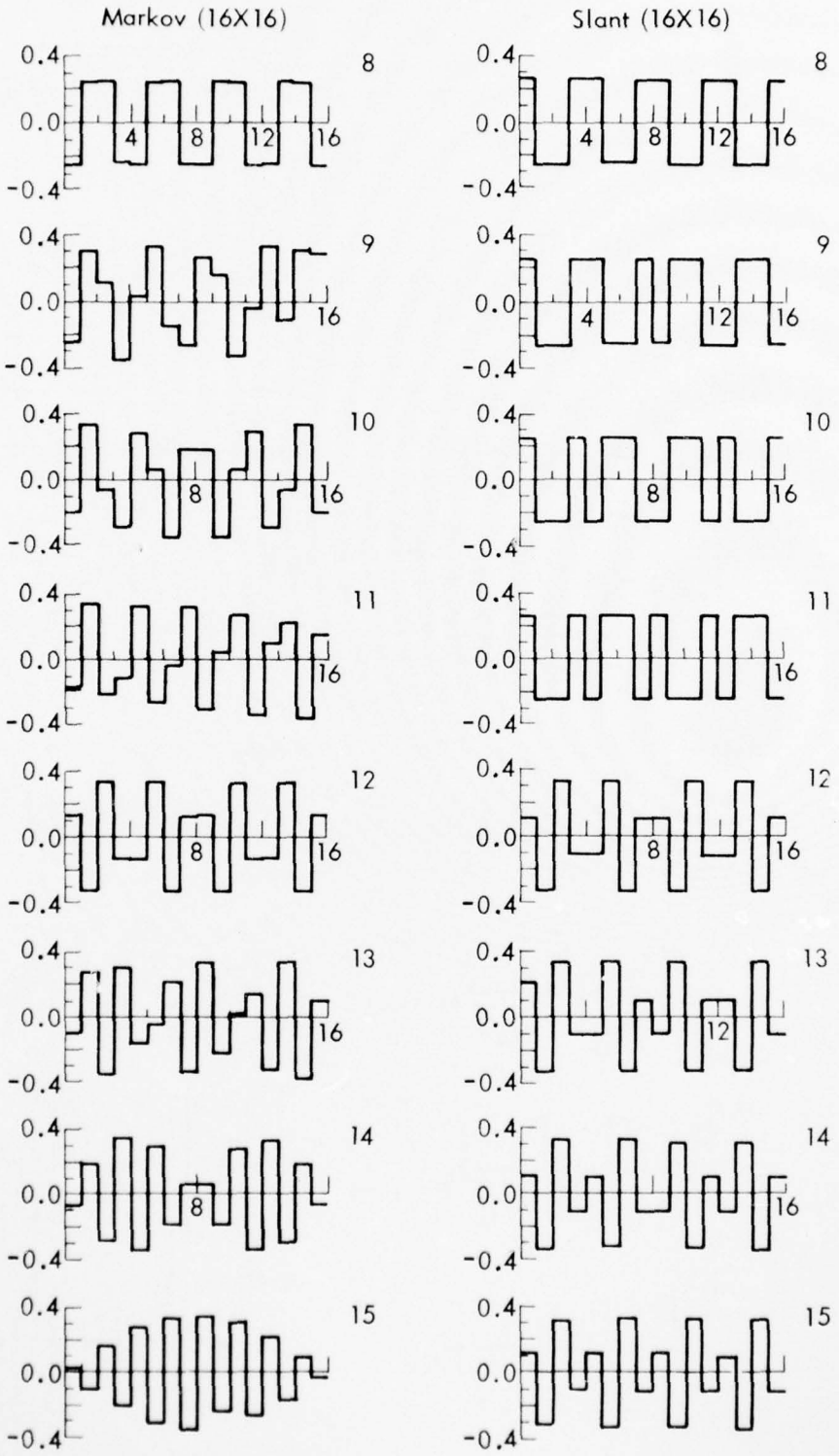


Figure 4.25b. Sequency Comparison

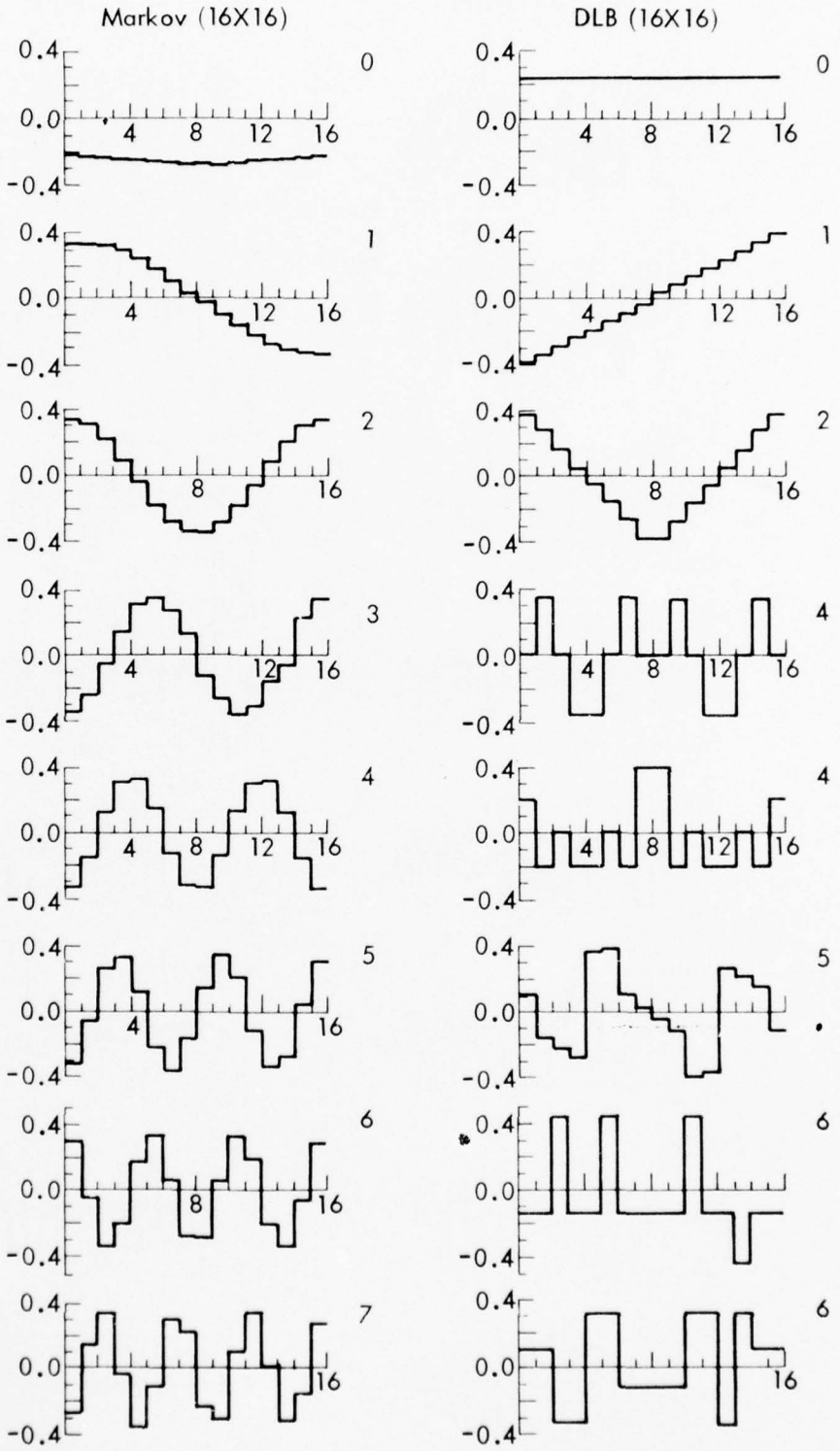


Figure 4.26a. Sequency Comparison

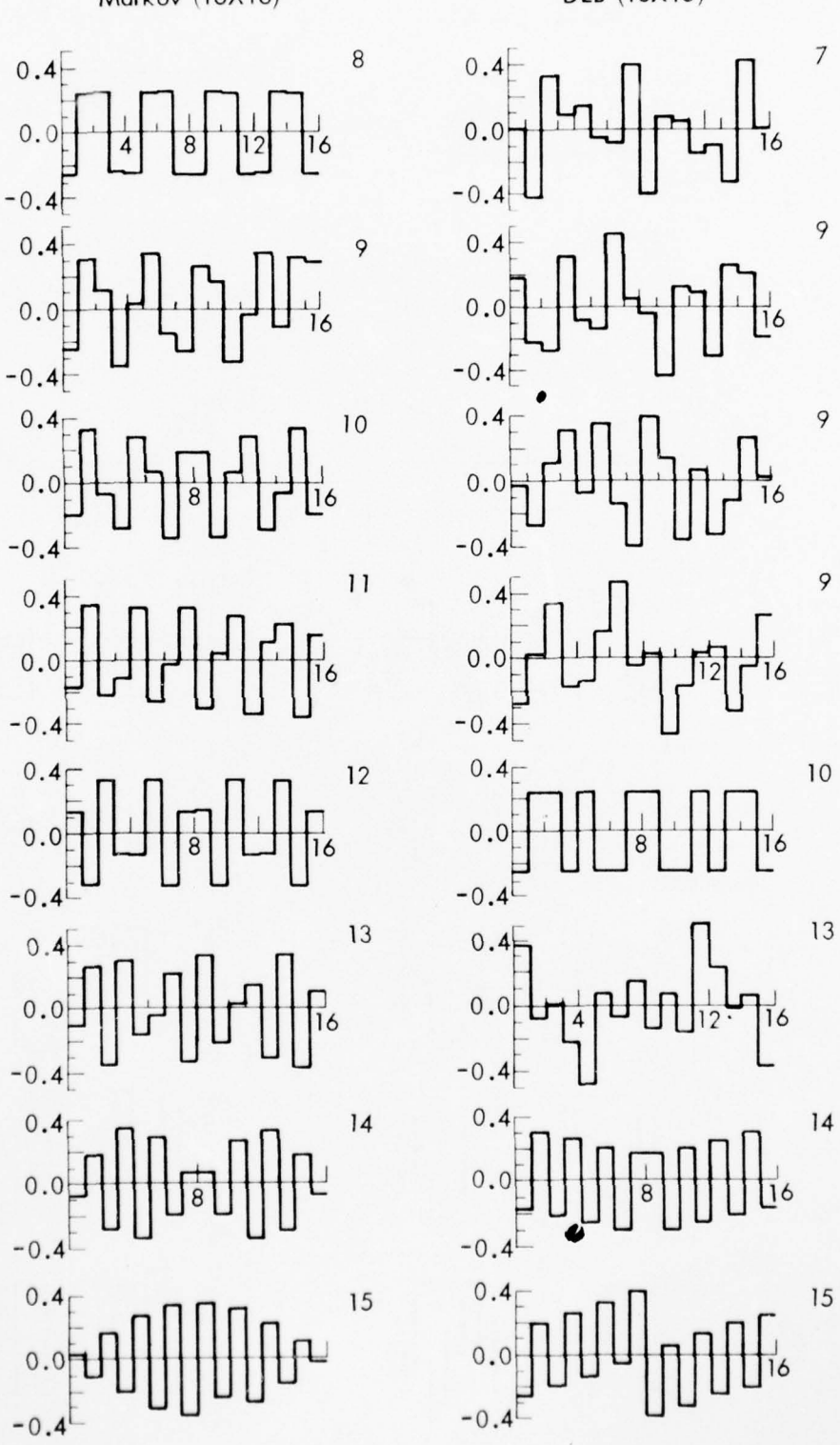


Figure 4.26b. Sequency Comparison

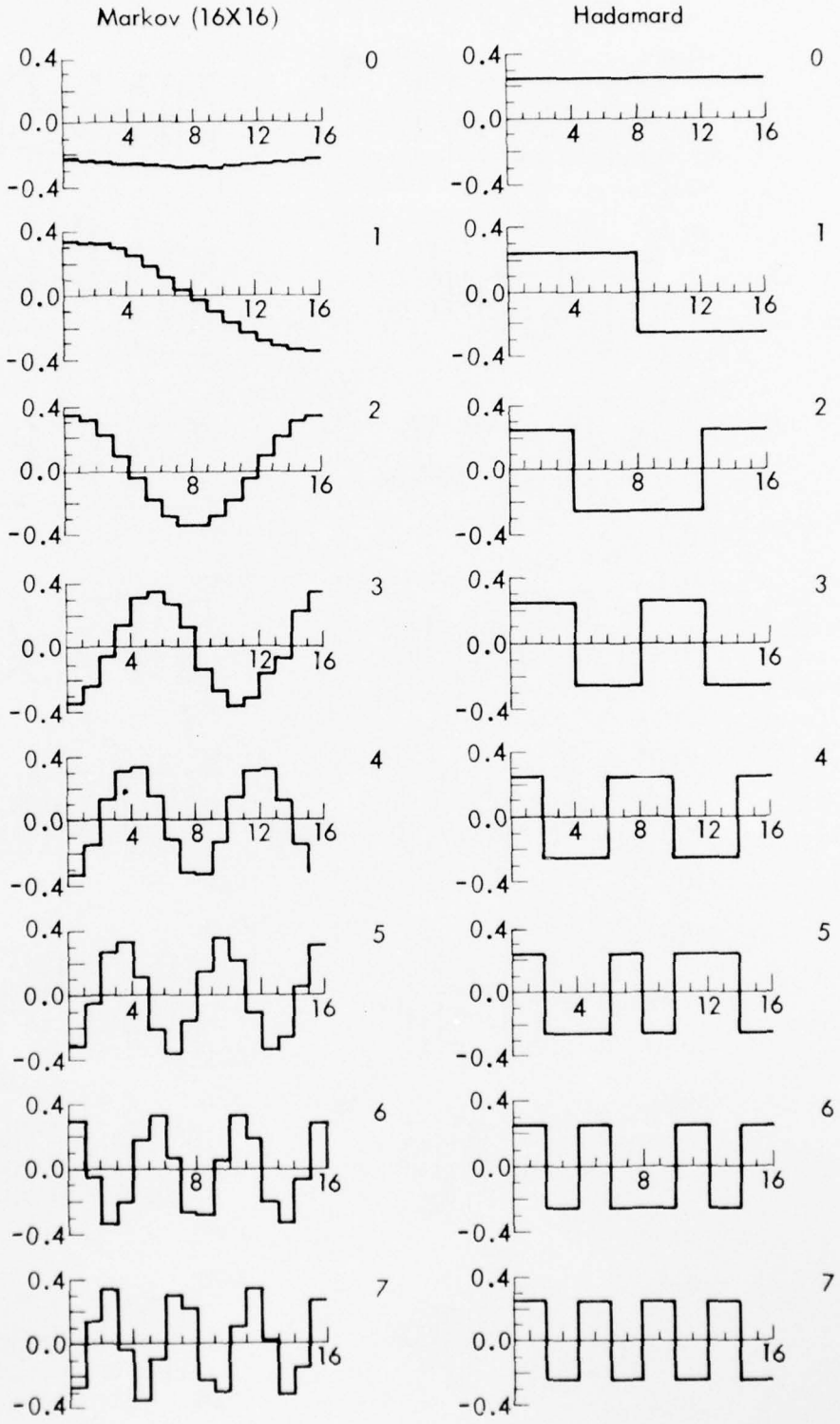


Figure 4.27a. Sequency Comparison

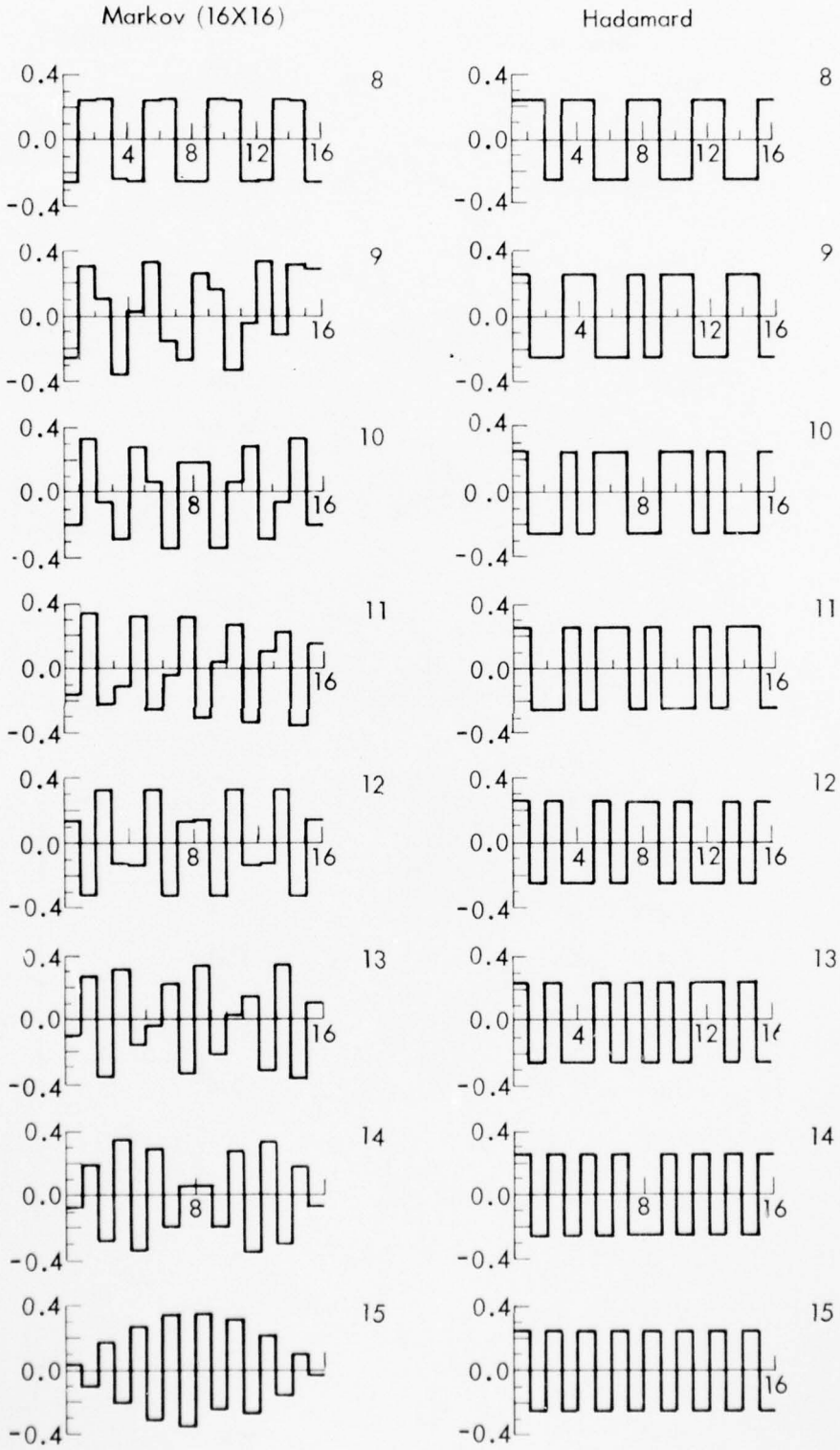


Figure 4.27b. Sequency Comparison

SECTION V

THE INTERFRAME PROCESS

1. INTRODUCTION

The previous chapters have dealt with the case of one mono-chromatic frame. To consider the system concept the multi-frame process must be considered. The achievement of large compression ratio's, in the order of 50:1, rely upon capitalizing on interframe compression. Relatively few research results have been published aside from theoretical results, which report on any but well known techniques for images in which there is motion. The latest approach discussed by Habibi [1] and further results, by the University of Southern California (SC) [2], use a three-dimensional transform technique. The difficulty here is the complexity of three-dimensional transforms and the storage required. In the more published DPCM approach, only one frame need be stored at a time and the implementation is relatively easy, however, the compression ratio is low.

In this chapter we report on a suggested combination of transform compression, DPCM, and frame rate reduction. It is this combination which evokes the necessary 50:1 compression for the system concept.

2. Channel Noise for the Discrete Cosine

In the selection of the Discrete Cosine transform as the "best" for the interframe study we must concern ourselves with its response to channel noise. To yield an idea of how this transform might perform an assumption of a Binary Symmetric channel with bit by bit transmission was made. Although not a complete analysis it may be used as an indicator of probable performance for given noise probabilities.

In any given transform, the variable length coding scheme (Chapter 3) assigns a fixed number of bits to a subimage. Under this constraint we can consider the sequence of Ø's and 1's as a sequence of independent Bernouilli trials. The probability of receiving (n-k) correct digits or exactly k erroneous digits is:

$$P\{k \text{ errors}\} = \binom{n}{n-k} p^{n-k} q^{k}$$

where:

$$q = 1 - p$$

If we consider an N-dimensional vector communication channel where the transmitter is defined by a set of M signal vectors, $\{S_i\}$, when message m_i is selected for transmission, vector S_i is transmitted.

$$S_i = (s_{i1}, s_{i2}, \ldots, s_{iN}), \quad i=0,1,\ldots,M-1$$

If we choose a convenient set of orthonormal functions $\{\phi(t)\}$ the signal may be defined in this signal space with N mutually perpendicular axis labeled $\phi_1, \phi_2, \ldots, \phi_N$. If $\bar{\phi}_j$ denotes the unit vector along the j^{th} axis, $j=1,2,\ldots,N$ then each N tuple above describes the vector

$$\bar{S}_i = S_{i1}\bar{\phi}_1 + S_{i2}\bar{\phi}_2 + \ldots + S_{iN}\bar{\phi}_N$$

If we treat our image data as $M = 2^N$ equally likely messages and locate these signals in an orthogonal space, i.e., vertices of an N dimensional hypercube, then Wolzencraft and Jacobs [3] prove that no error is made if the noise vector $n_i < \frac{d}{2}$ where d is the distance between vertices, and in fact $\frac{d}{2}$ is the decision boundary. Now for the assumption of white additive noise, a constant term $\frac{N_o}{2}$ with a variance $\sigma^2 = R(\tau) = \frac{N_o}{2}\ \delta(\tau)$ for zero mean is made. Thus the joint

density function p_n is

$$p_n(\alpha) = \frac{1}{(\pi N_o)^{N/2}} e^{-|\alpha|^2/N_o}$$

The probability of error is then:

$$p[E|m_1] = \int_{\frac{d/2}{\sqrt{N_o/2}}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\gamma^2/2} d_\gamma = \phi\left(\frac{d}{\sqrt{2N_o}}\right)$$

where ϕ is the erf function defined by:

$$\phi(\alpha) = \frac{1}{\sqrt{2\pi}} \int_\alpha^\infty e^{-\gamma^2/2} d\gamma$$

The error probability for this study is a given, and takes on the values of $p = 10^{-2}, 10^{-3}, 10^{-4}$. If we choose a random variable V to represent the number of erroneous digits in a message then V takes on values $\mu = 0,1,2,\ldots,n$ with probabilities:

$$P\{V = \mu\} = \binom{n}{n-\mu} p^{n-\mu} q^\mu$$

and the average value is:

$$E\{V\} = \sum_{\mu=0}^{n} \mu P\{V = \mu\} = nq$$

Stated simply in each sequence of n binary digits we can expect $|nq|$ to be altered by noise, on the average. The simulation, therefore, generates an average of $|nq|$ alterations by adding white noise to the coded bit assignments.

The Discrete Cosine Transform and optimum bit allocation programs were combined to investigate the noise effects. Each sub-image was treated as a random message in a binary symmetric channel. The white noise generation from the computer made it possible to randomly change message bits at the

given threshold corresponding to the probability of error.

The results indicated a steep rise in the RMS error of the image as a function of the error probabilities (Figure 5.1). This steep increase in error can be explained by the way the system is simulated. If eight bits, for example, are assigned to a component by the coding algorithm there is a possibility of 256 "Max" quantized values being used. These estimated coefficients are not actually subjected to the error. Instead, an index is transmitted for each component indicating which level of the 256 should be used. A small change in the predominant bit of the index could make a gross change in the coefficient level used. The RMS error of the reconstructed picture is, therefore, a function of the simulation used and should not be used as conclusive evidence of the effect of errors in the channel.

Reasonable specification limits in terms of error probabilities are 10^{-5} to 10^{-6}. Although not requested, the system was simulated at 10^{-5} probability of error. A comparison of RMS error with and without channel noise resulted in the following:

Avg. bits/pel	RMS No Noise	RMS White Noise $p = 10^{-5}$
1 bit/pel	3.6503	3.6504
.75 bit/pel	3.9727	3.9749
.5 bit/pel	4.43664	4.4365

It may be concluded therefore that except for error probabilities higher than 10^{-5}, this system coding is reasonably accurate.

3. DPCM

A common method for transmitting data and achieving moderate compressions is Differential Pulse Code Modulation (DPCM). In this approach, the differences of successive

AD-A039 761 KANSAS UNIV/CENTER FOR RESEARCH INC LAWRENCE F/G 17/2
VIDEO BANDWIDTH COMPRESSION.(U)
FEB 77 N C GRISWOLD, R M HARALICK F33615-74-C-1093
UNCLASSIFIED 257-4 AFAL-TR-76-102 NL

3 OF 3
AD
A039761

END

DATE
FILMED
6-77

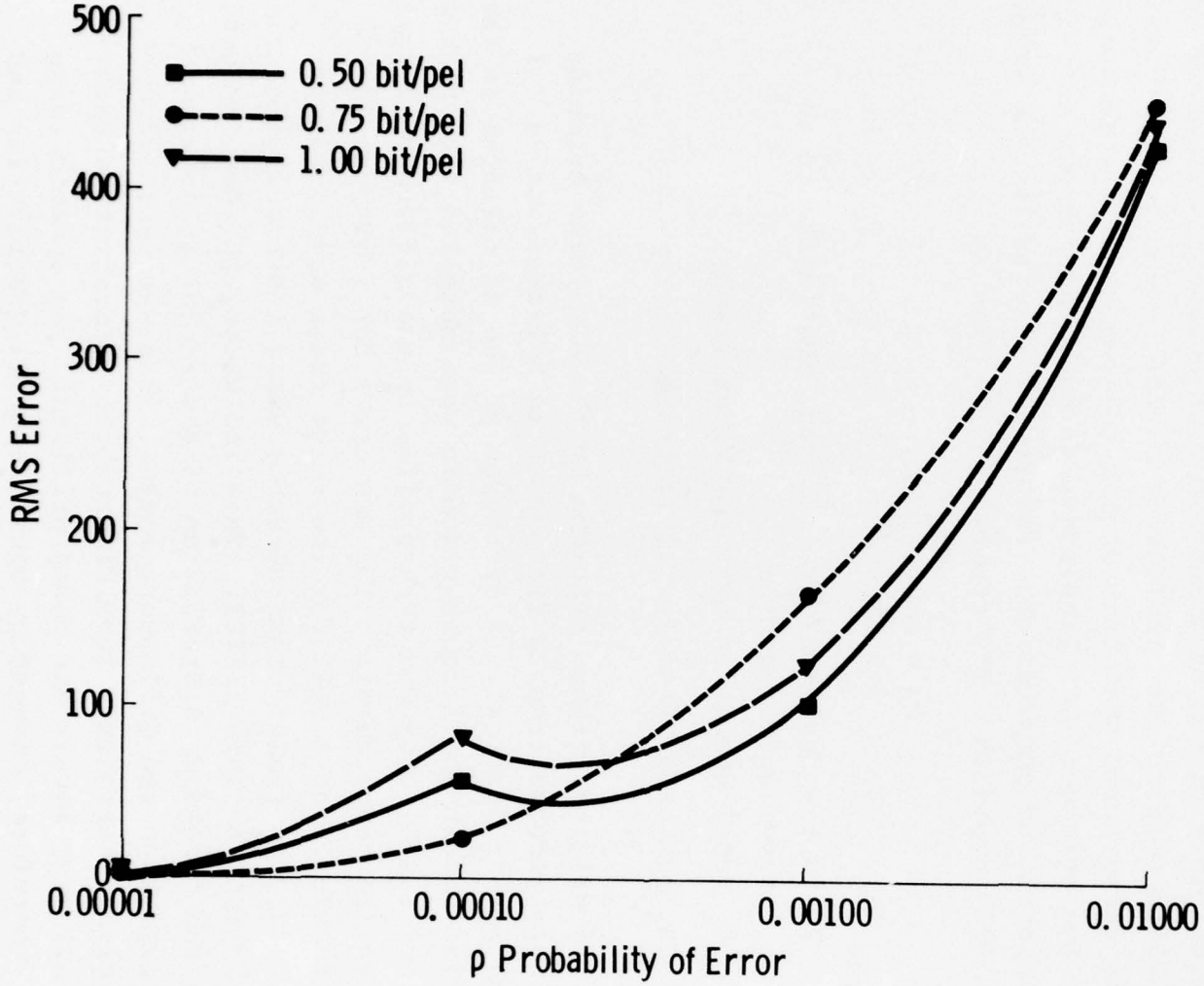


Figure 5.1. RMS Error as a function of error probability.

signals samples are transmitted rather than the signals themselves. This has been adapted for use in the transform domain. This method of transform/DPCM encodes differences in transform coefficients instead of picture element amplitudes. It is anticipated that this should be less sensitive to noise than the straight DPCM predictive schemes and achieve an additional 1.5 compression. Figure 5.2 illustrates the transform DPCM approach. The compression is given by:

$$CR = \frac{KN_t}{N_t + (K-1)N_i}$$

where:

- K is the number of frames being considered to an error specification
- N_t total number of bits for first frame
- N_i number of bits for (K-1) frames

The original frames supplied by the Air Force Avionics Lab were reduced from 24 frames/sec to 8 frames/sec (a 3:1 frame rate reduction). A histogram of the differences in the spatial domain and frequency domain were compared (see Figures 5.3 and 5.4). DPCM was then applied to the difference images in the frequency domain. The RMS error for 1 bit/pel (6:1 compression) was 1.0012 for the first frame and 1.5033 for the predicted frame. The RMS error using 1 bit/pel without prediction was only 1.2202. This indicated, on these images, that the frequency distribution of the actual image was narrower than the difference image. This was verified by additional histograms (see Figure 5.5). This infers that not all gray levels are being utilized. The dynamic range was, therefore, changed by equal interval quantization and the resulting histogram was wider than previously (Figure 5.6) and the RMS errors were reduced to 1.09 for predicted value. DPCM could then be used efficiently. Coding the differences

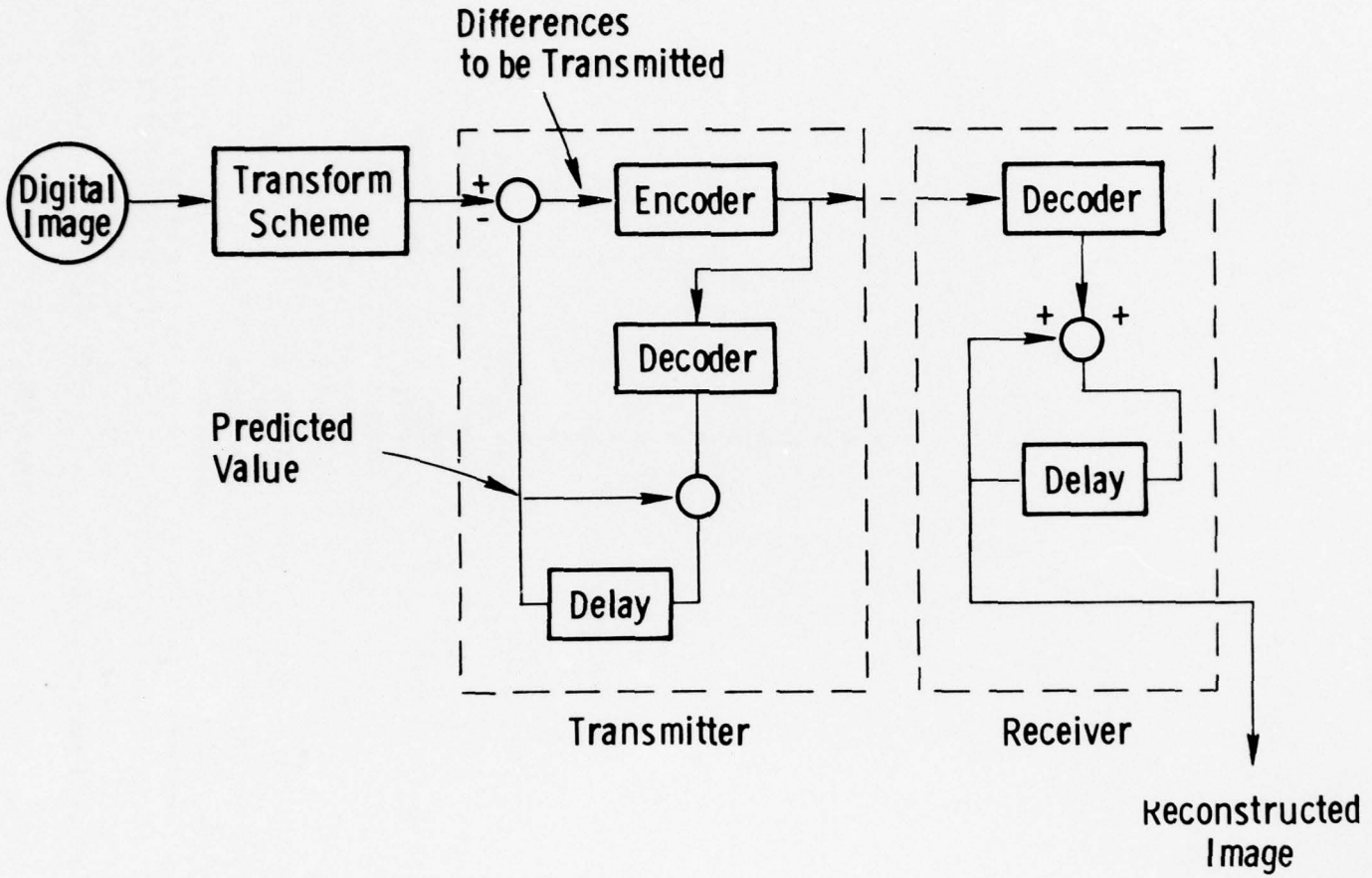

Differences
to be Transmitted
Digital
Image
Transform
Scheme
+
-
Encoder
Decoder
Predicted
Value
Delay
Transmitter
Decoder
+
+
Delay
Receiver
Reconstructed
Image

Figure 5.2.

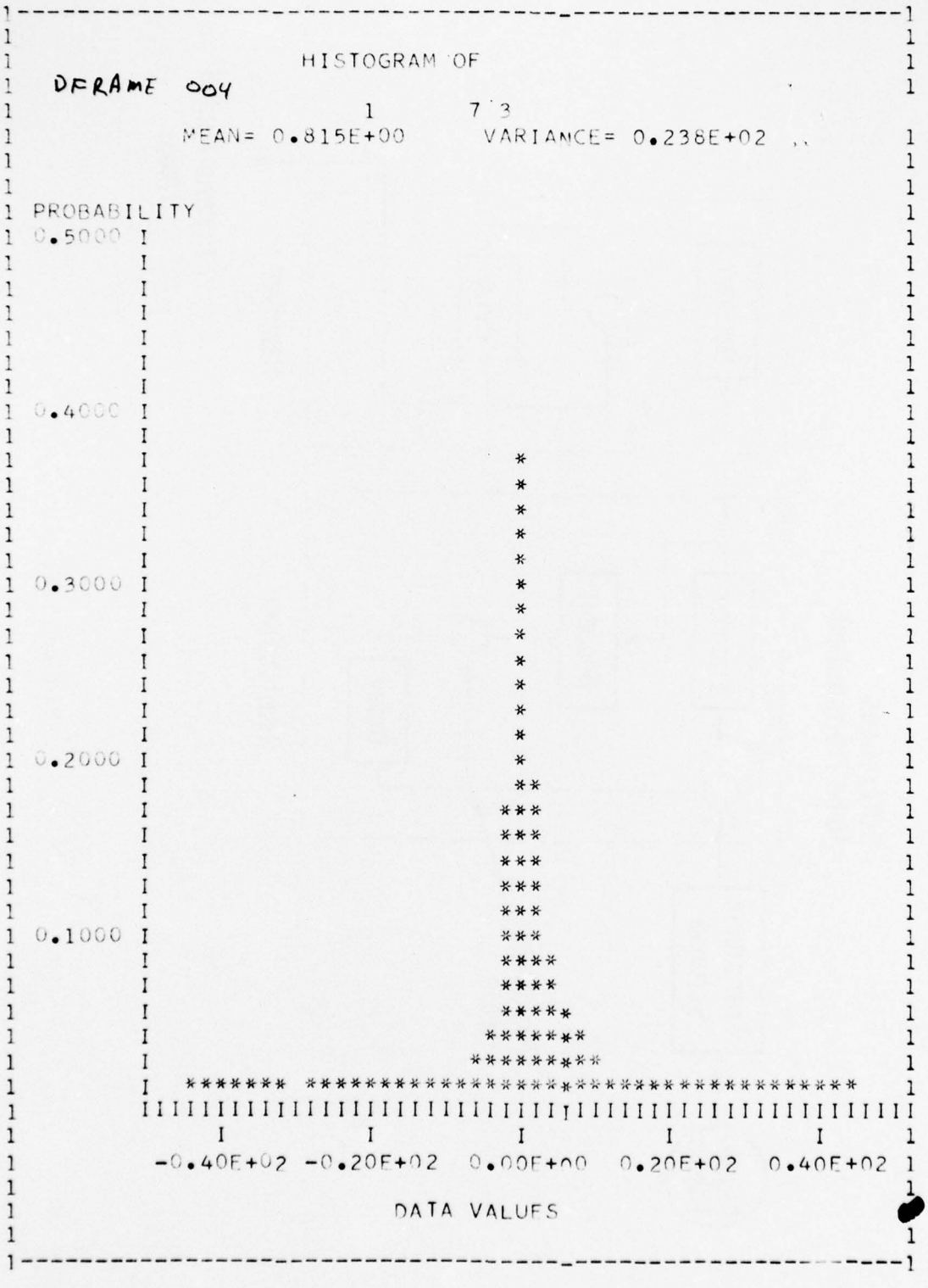


Figure 5.3. Histogram of Spatial Differences for DPCM

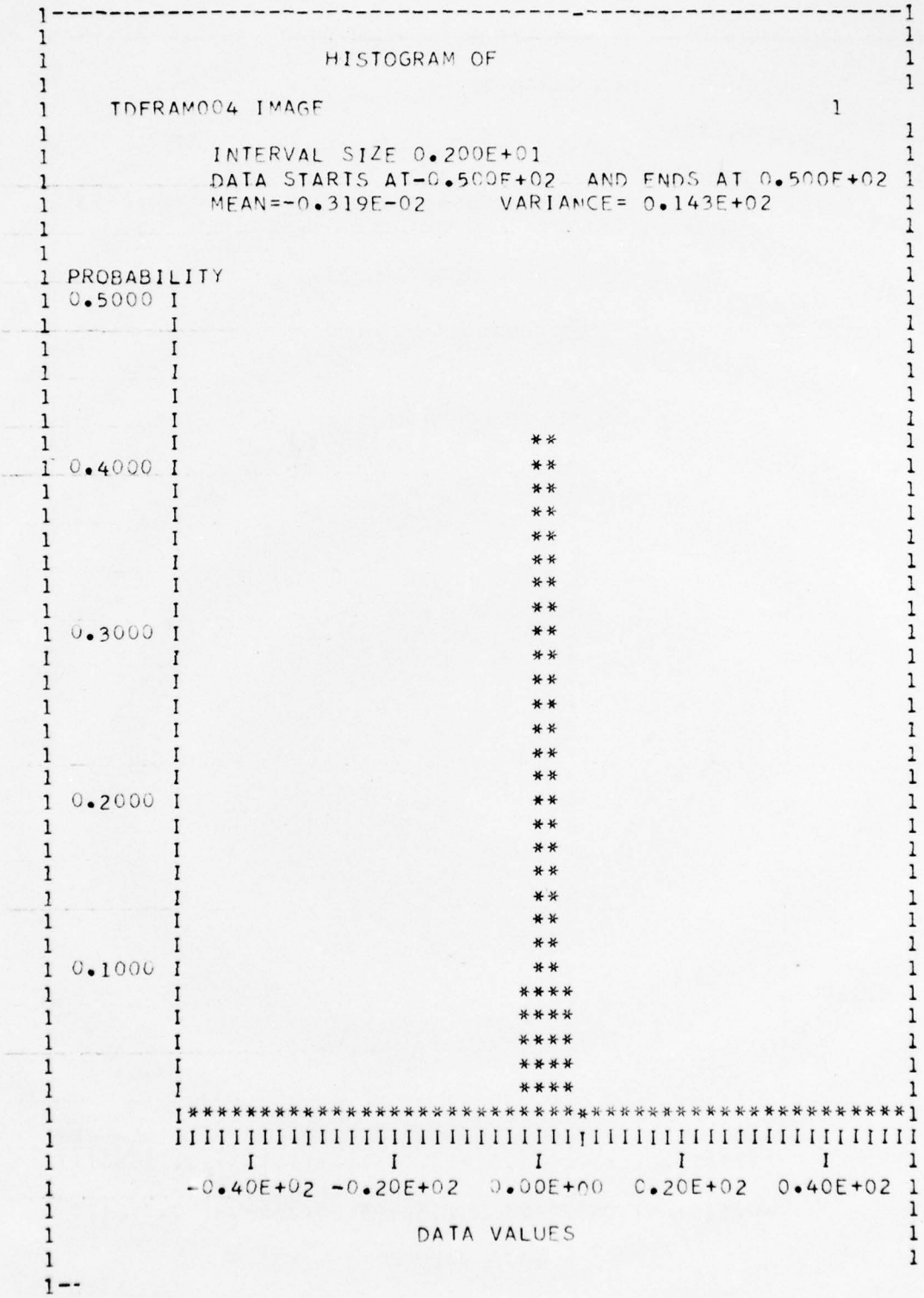


Figure 5.4. Histogram of Frequency Differences for DPCM

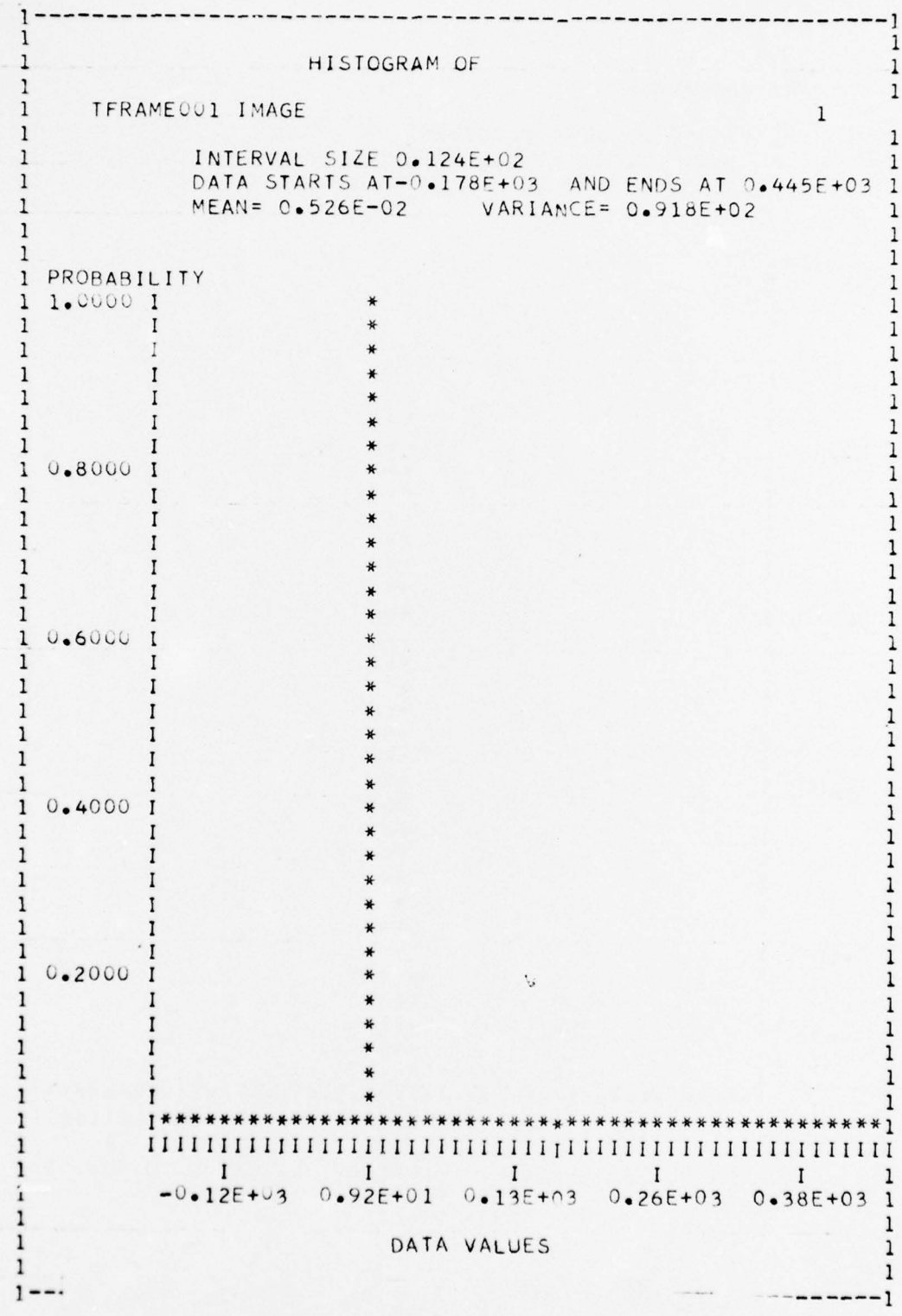


Figure 5.5. Histogram of transformed original

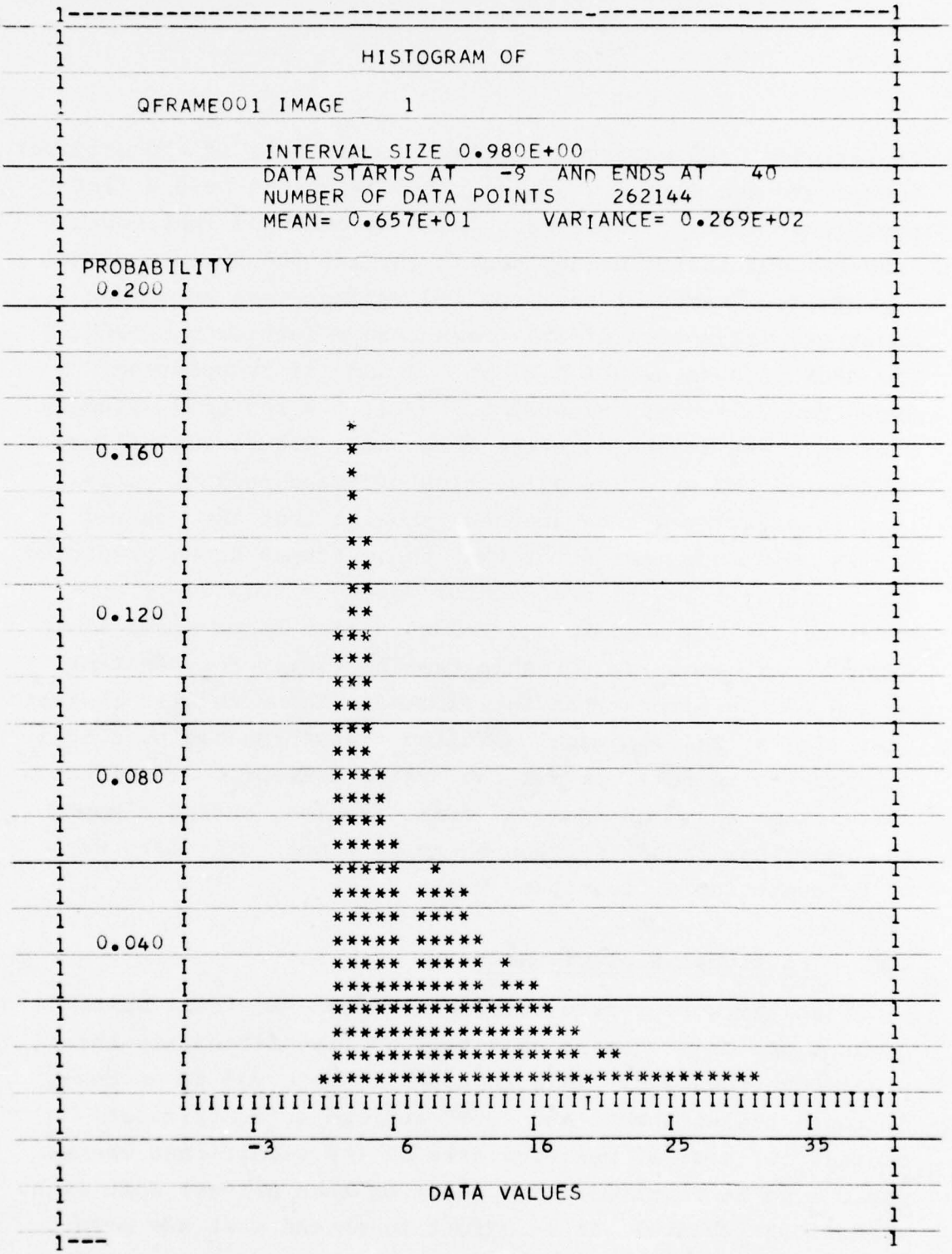


Figure 5.6. Histogram of original with equal interval quantization to 32 levels.

is efficient and produces savings in code only if the original images are not flat. These supplied images do have a flat spatial property. Therefore, it is recommended that equal interval quantizing be applied to utilize the dynamic range.

Twelve frames of the supplied imagery were processed under the assumption of 3:1 frame rate reduction and DPCM. The DPCM process used a C.R. of 1.5 and K=4 or updating every fourth frame. Figures 5.7, 5.8, 5.9 and 5.10 visually compare these frames with the original. Figure 5.11 illustrates the RMS error as a function of frame number. This plot is actually a time sequence showing that the updated frames have much less error than those frames being predicted. The original frame to frame error which is relatively constant may be compared to the reconstructed frame error as in Figure 5.12. The relationship remains linear for constant motion of the aircraft as anticipated. These results indicate that with a 12:1 compression in the transform domain, a 50:1 compression is feasible for the system concept. Since the correlation based on frame-to-frame is high, indeed almost the identical image, perhaps there is a more efficient way of transmitting differences.

4. Conditional Replenishment

Since there is little change on a frame-to-frame basis perhaps the only changes that need be transmitted are the significant changes. This approach is referred to as conditional replenishment and would achieve more efficient coding. Of course, position data of the significant change would also be required. The question then arises; what is a significant change? In an effort to extend what has been reported so far in this project a suggestion for a significant change criteria is made.

Original Quantized

1

2

3

Reconstructed

Comparison of Interframe Sequences

Figure 5.7

Original Quantized

4

5

6

Reconstructed

Comparison of Interframe Sequences

Figure 5.8

Orfginal Quantized

7

8

9

Reconstructed

Comparison of Interframe Sequences

Figure 5.9

Original Quantized

10 11 12

Reconstructed

Comparison of Interframe Sequences

Figure 5.10

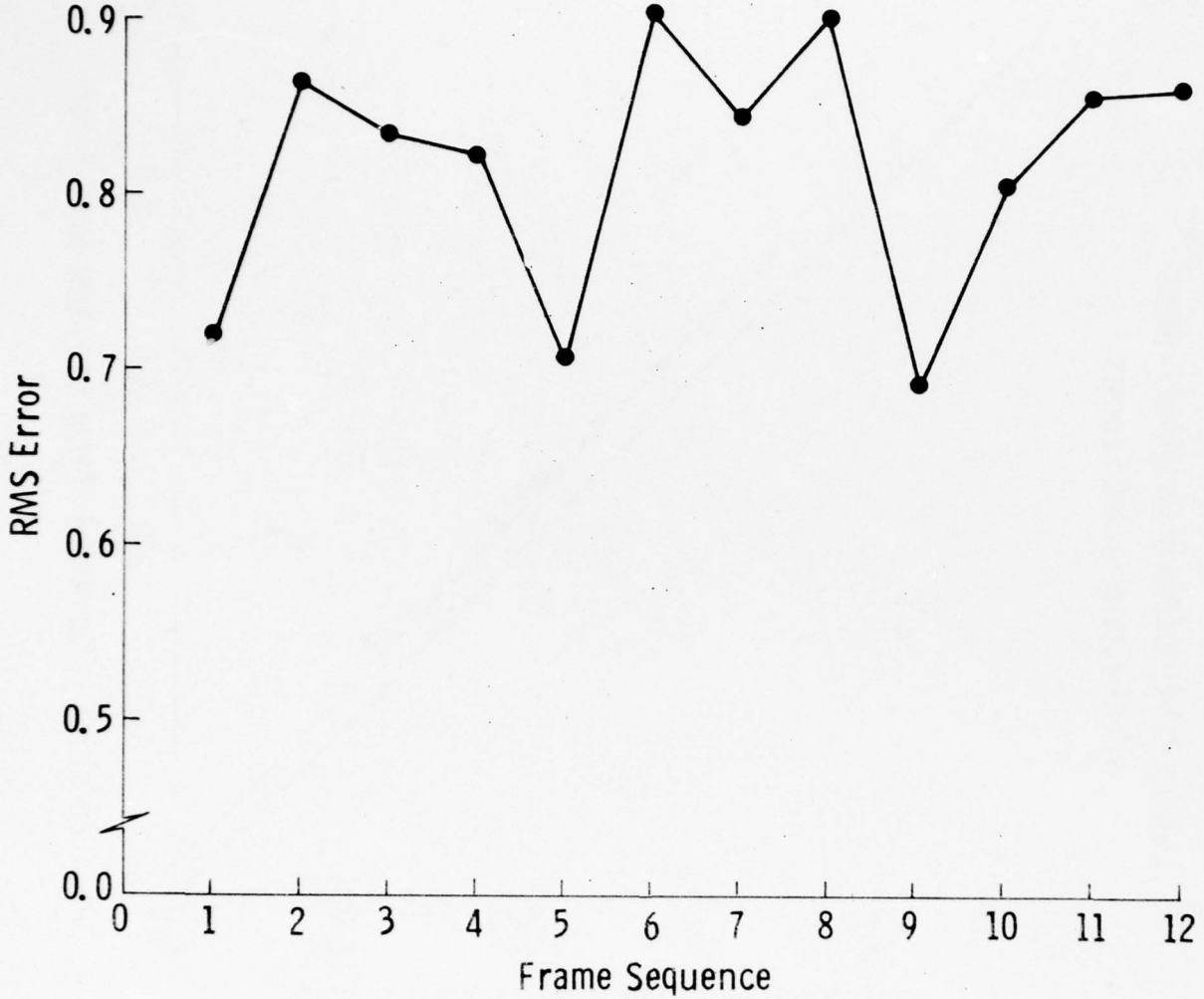


Figure 5.11. RMS Error as a Function of Frame Number

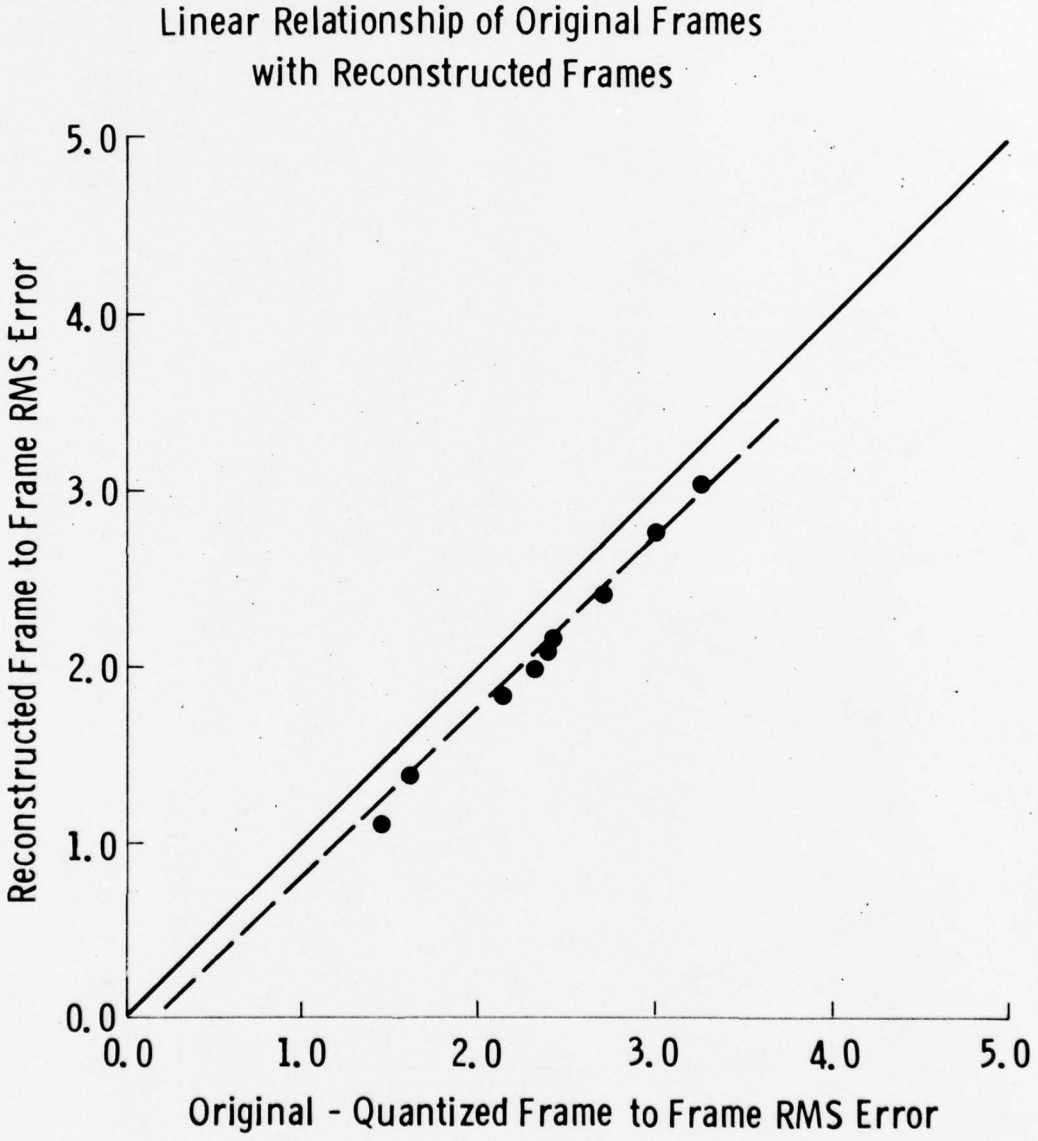

Linear Relationship of Original Frames
with Reconstructed Frames
Reconstructed Frame to Frame RMS Error
5.0
4.0
3.0
2.0
1.0
0.0
0.0
1.0
2.0
3.0
4.0
5.0
Original - Quantized Frame to Frame RMS Error

Figure 5.12.

a. Criteria for Significant Change by Correlation Relations

Hypothesis: A figure of merit exists which relates change in a frame-to-frame basis in terms of correlation. Define the set $x_1, x_2, \ldots, x_p$ to be random variables of the observation vector of any sub-image. Let the observation vector be designated by $\overline{X}_m$ where m denotes the m^{th} frame and the range of m is 1...M. Capital M is the total number or frames.

Then:

$$\overline{X}_m = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_p \end{pmatrix} \qquad p\text{x}1$$

The covariance of X_m (the observation vector, if the set of random variables $\{x_1, x_2, \ldots, x_p\}$ has zero mean is:

$$\text{covariance of the } m^{th} \text{ frame} = \text{Cov}(\overline{X}_m) = E\{\overline{X}_m \overline{X}_m{}^T\} = \frac{1}{N} \sum_{k=1}^{N} \overline{X}_{mk} \overline{X}_{mk}{}^T$$

where N is the total number of subimages.
Let the preceeding frame be designated by an observation vector $X_{(m-1)}$ the covariance of this frame is

$$\text{covariance of the } (M-1)^{th} \text{ frame} = \text{Cov}(\overline{X}_{m-1}) = E\{\overline{X}_{m-1} \overline{X}_{m-1}^T\} = \frac{1}{N} \sum_{k=1}^{N} \overline{X}_{(m-1)K} \overline{X}^T (\text{illegible}$$

Now define the sequence observation vector as:

$$S = \begin{pmatrix} \bar{X}_m \\ \bar{X}_{m-1} \\ \vdots \\ X_1 \end{pmatrix} \quad Mx1$$

The covariance of sequence of frames is then:

$$Cov(s) = E\{S\ S^T\}$$

$$E = \left\{ \begin{pmatrix} \bar{X}_m \\ \bar{X}_{m-1} \\ \vdots \\ \bar{X}_{m-\ell} \end{pmatrix} \begin{pmatrix} \bar{X}_m^T & \bar{X}_{m-1}^T & \cdots & \bar{X}_{m-\ell}^T \end{pmatrix} \right\}$$

$$= \begin{pmatrix} E\{\bar{X}_m \bar{X}_m^T\} & E\{\bar{X}_m \bar{X}_{m-1}^T\} & \cdots & E\{\bar{X}_m \bar{X}_{m-\ell}^T\} \\ E\{\bar{X}_{m-1} \bar{X}_m^T\} & E\{\bar{X}_{m-1} \bar{X}_{m-1}^T\} & \cdots & E\{\bar{X}_{m-1} \bar{X}_{m-\ell}^T\} \\ \vdots & \vdots & \cdots & \vdots \\ E\{\bar{X}_{m-\ell} \bar{X}_m^T\} & E\{\bar{X}_{m-\ell} \bar{X}_{m-1}^T\} & \cdots & E\{\bar{X}_{m-\ell} \bar{X}_{m-\ell}^T\} \end{pmatrix}$$

for case of two frames =

$$\left(\begin{array}{c|c} \sigma^2_{11(m,m)} \cdots \sigma^2_{pp(m,m)} & \sigma^2_{11(m,m-1)} \cdots \sigma^2_{pp(m,m-1)} \\ \hline \sigma^2_{11(m-1,m)} \cdots \sigma^2_{pp(m-1,m)} & \sigma^2_{(m-1)(m-1)} \cdots \sigma^2_{pp(m-1,m-1)} \end{array}\right)$$

Three important properties of this covariance of time varying errors are:

a. The random processes giving rise to the error vectors may or may not be stationary. If the second order statistics are changing in time this information will be clearly conveyed by the successive total covariance matrice.

b. The matrices $E\{\overline{X}_{M-i}, \overline{X}^T_{M-j}\}$ for $i \neq j$ may or may not be null. These matrices form the off diagonal elements of covariance $(\overline{S}\ \overline{S}^T)$, so if observation errors of any one frame with any other frame are uncorrelated these terms will be null. If correlation exists some or all may be non null. [Stage wise correlation].

c. The main diagonal blocks (note single frame subscript) may or may not be uncorrelated. If they are uncorrelated then we have locally non-correlation. As an example:

$$\text{Cov}\ (\bar{S}) = \begin{pmatrix} \alpha^2 & & & \\ & \alpha^2 & & \\ & & \ddots & \\ & & & \alpha^2 \end{pmatrix}$$

where α is a constant then:

we would have a. second order statistics stationary

b. errors are stagewise uncorrelated

c. errors are locally uncorrelated.

If, however,

$$\text{Cov}(S) = \left(\begin{array}{cc|cc} \theta^n & \theta^n & & \\ \theta^n & 2\theta^n & \multicolumn{2}{c}{\raisebox{1.5ex}{$\emptyset$}} \\ \hline & & \theta^{n-1} & \theta^{n-1} \\ \multicolumn{2}{c|}{\raisebox{1.5ex}{$\emptyset$}} & \theta^{n-1} & \theta^{n-1} \end{array}\right)$$

n,n-1 are two consecutive frames.

then we have (where θ is a real constant)

a. non-stationary

b. stagewise uncorrelated

c. locally correlated

Therefore since the frame-to-frame correlation is higher we would expect to see more locally uncorrelated evidence and high stagewise correlations. Unless a significant change occurs, then stagewise correlation should be less.

So define F a figure of merit

$$F(m,m-1) = \frac{\text{trace } \ddagger_{m-1,m}}{\sqrt{\text{trace } \ddagger_{m-1,m-1} \text{ trace } \ddagger_{mm}}}$$

where m,m-1 denotes two frames.

$\ddagger$ covariance matrix is partitioned form. See Figure 5.13.

This process may also be applied to error images which are made up of the differences of any two consecutive frames.

If we consider any one frame Y_m we can estimate the next frame by a prediction filter

$$\hat{Y}_{m+1} = W(p)Y_m$$

where $\hat{Y}_{m+1}$ is the predicted observation

$W(p)$ is a transition matrix with p-step prediction. In this example p = 1.

This filter would also operate on the error images to predict the next error image. If E_ℓ represents the difference image between frame m and m-1 then $\hat{E}_{\ell+1} = W(1)E_\ell$ is the estimate of the next error image. W(o) matrix = a zero order predictor would be

$$W(o) = \begin{bmatrix} 1 & & & \emptyset \\ & 1 & & \\ & & 1 & \\ & & & 1 \\ & & & & \ddots \\ \emptyset & & & & & 1 \end{bmatrix} = I$$

so that $\hat{Y}_m = W(o)Y_m$. Each of the new covariances may be defined by covariance (S) = $E\{S\ S^T\}$ and the estimated

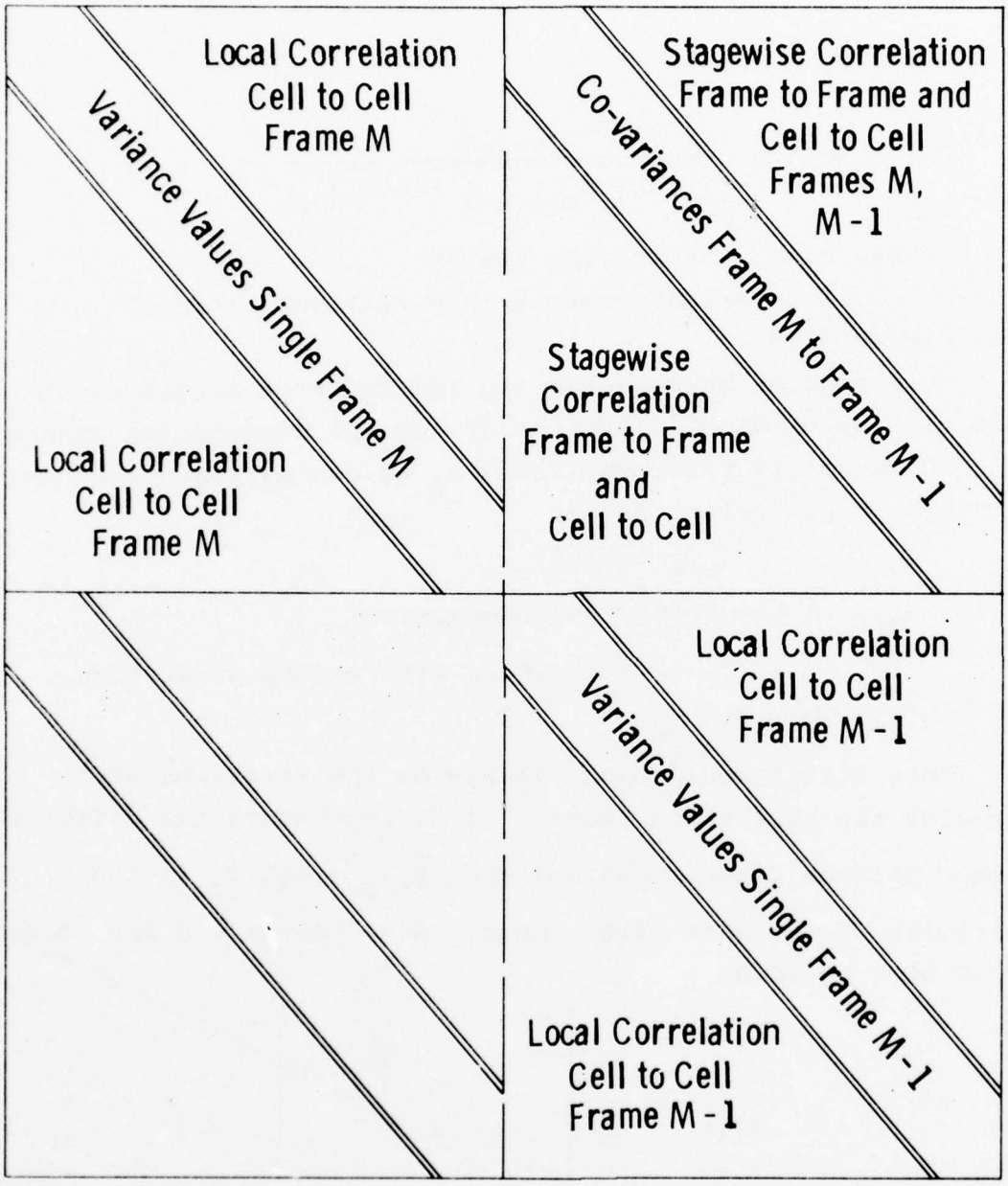


Figure 5.13. Co-variance matrix for two successive frames illustrating stagewise correlation, local correlation.

covariance (S) = $W(p)E\{S\ S^T\}W^T(p)$. Therefore in conclusion as each frame is transmitted a several step prediction can be used to predict the next few frames. This will smooth the image transitions, and save bandwidth by reducing the number of frames transmitted. Further the figure of merit for additional frames can be estimated by the prediction process.

SECTION VI

CONCLUSIONS AND RECOMMENDATIONS

1. CONCLUSIONS

It may be concluded from the data submitted within this report that the Fast Karhunen Loeve more closely approximates the optimum Principal Components than any other fast transform. This transform also exhibits an adaptive nature since the eigenvectors used for the basis set change with each image. If a class of images which have nearly the same eigenvectors is found the implementation would be as rapid as any other fast transform.

The Discrete Cosine was chosen as the "best" transform because of its small error and ease of implementation. Hardware implementation of this transform may be feasible. The Discrete Linear Basis with the (8·2·8·2) direct product implementation is nearly as accurate as the Discrete Cosine and in fact more feasible for hardware implementation since it can be accomplished with integer arithmetic. The small error would be less significant than the savings in storage and running time.

In terms of the optimum bit allocation with the Discrete Cosine it appears that sufficient operation could be accomplished with the noisy channel if the noise is held to less than 10^{-5} probability of error for a selected transmission rate.

From a system point of view it is feasible to compress the images in terms of 50:1 with reasonable errors. If additional errors may be tolerated, or flicker with frame rate reduction in the order of 5 to 10:1. The implementation of the hybrid Discrete Cosine/DPCM seems reasonable compared to the required storage for three dimensional transform approaches.

2. RECOMMENDATIONS

In the process of this investigation several unresolved problems exist which it is recommended be pursued. The Fast K-L transform basis set is image dependent. The question must then be asked how much error exists in contiguous imagery if the basis set is not changed for each frame. Further how many frames could be used or is this particular transform possible for an entire class of imagery such as desert areas or forest areas where the background does not change significantly.

Since the Fast K-L transform indicates that the optimum number of layers for a fast transform is two for the direct product implementation, why was the Discrete Linear Basis using two layers poorer than the direct product of 8 and 2 resulting in four layers? The question should be pursued in the basis of the optimum generation of a 16 x 16 basis set and investigation of why the sequency property is lost. If this happens in general then what ordering of the basis vectors should be utilized.

Further improvement of images under compression may be possible by a permutting process. A brief look at this type implementation revealed it would be necessary to transmit the mean of each sub-image rather than the mean of the entire image. Time did not allow a complete analysis of this approach.

Another unresolved problem is the extension of Haralick's storage savings Discrete Cosine implementation efficiently to two dimensions. The present method requires a subroutine call to a Fast Fourier Routine four times which increases the running time. An efficient method by calling the Fast Fourier routing only once for the forward and inverse transforms seems reasonable but has not been found.

Further, recommendations must be made concerning the conditional replenishment. With the tools now available, a sequence of frames with much more activity should be examined. Also built into our suggested significant change criterion is an overall averaging effect. This could possibly be examined for different size subimages or expanded so it is possible to decide what subimages should be changed, not just frames.

APPENDIX I

IMAGE DATA COMPRESSION: THE INCOMPLETE FAST TRANSFORM

Robert M. Haralick & Norman Griswold

Remote Sensing Laboratory
University of Kansas
Lawrence, Kansas 66045

ABSTRACT

One frequently used image data compression method is based on transform coding. In this paper we show that transform coding of image data really consists of applying a fast transform in an incomplete fashion. In terms of RMS error, the best transform is the principal components (Karhunen Loéve) one. We show that under certain conditions there exists a fast principal components transform.

INTRODUCTION

One of the first pattern recognition tasks is the preprocessing of data to normalize and/or reduce dimensionality. Data compression is a natural place to look for information preserving procedures which reduce the number of data points or dimensions. One frequently used data compression method is that based on transform coding using one of the fast transforms such as FFT, FHT or DLB. In this note we illustrate that transform coding of image data really consists of applying a fast transform in an incomplete fashion. In addition we give a semi fast algorithm for the KL transform of an isotropic 4 x 4 image and discuss the additional assumptions for the fast transform.

Figure 1 illustrates the general idea of transform coding. The image is partitioned into equal sized rectangular subimages, say M rows by N columns each, a fast forward transformation is done on each subimage, and compression is achieved by selecting only those transformed components having relatively high energy (the non-selected components are set to zero). A reconstruction of the compressed image is achieved by applying an inverse transformation, subimage by subimage, using a zero value for the non-selected transformed components. From our perspective of the incomplete fast transform, the key details of the transform coding technique are the partitioning of the image into subimages and the application of a fast transform on each subimage.

The fast transform is based on a direct product of vectors concept (see Good, 1971; Haralick & Shanmugam, 1973). Let x be an Mx 1 vector

$$x' = (x_1, \ldots, x_M)$$

and y be an Nx1 vector

$$y' = (y_1, \ldots y_N)$$

Then the direct product $x \cdot y$ of x with y is an NMx1 vector and can be defined by

$$(x\ y)' = (x_1y_1, x_1y_2, \ldots, x_1y_N, \ldots, x_My_1, \ldots, x_My_N)$$

To see how this relates to the fast transform, let

$$B_1 = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

and

$$B_2 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} \right\}$$

be two sets of basis vectors. B_1 is the 2 x 1 set of Hadamand vectors. B_2 is a 3 x 1 set of discrete lineas basis (DLB) vectors. Consider the set B_3 of vectors which results when we take the direct product of each vector in B_1 with each vector in B_2:

$$B_3 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -1 \\ 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 1 \\ 1 \\ -2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -1 \\ -1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 1 \\ -1 \\ 2 \\ -1 \end{pmatrix} \right\}.$$

There results a 6 x 1 set of discrete lineas basis vectors. Furthermore, as shown in figure 2, there exists a fast way to transfrom any vector x by the basis vectors in B_3. The reader may verify that the 4 x 1 Hadamand transform is defined by the direct product of B_1 with B_1. The form of the fast Fourier transform and the Slant transform are really just like the form simple fast transform shown in figure 2 but with minor variations. In the FFT, before most of the outputs of one layer can proceed to the next layer, they are multiplied by a complex number, the twiddle factor. In the Slant transform, before two of the outputs of one layer can proceed to the next layer, they are put through a 2 x 2 transformation. The FFT and Slant implementation are illustrated in figures 3 and 4 respectively.

Figure 5 shows a simple 4 x 4 image which we want to transform with a two-dimensional FFT. The definition of the general two-dimensional Fourier transform is given by

$$\bar{x}(m,n) = \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} e^{-2j\pi(\ell m + kn)/N} x(\ell,k) . \qquad (1)$$

Rearranging the summation,

$$\bar{x}(m,n) = \sum_{k=0}^{N-1} e^{-2j\pi kn/N} \left[\sum_{\ell=0}^{N-1} e^{-2j\pi \ell m/N} x(\ell,k) \right]$$

Thus we see that to do a two dimensional transformation we need only do a one-dimensional transformation for each row of x and after all rows have been transformed we do a one-dimensional transformation for each column. The two dimensional transformation on the N^2 components of x has no more than twice the number of operations as the corresponding one dimensional transformation on the N rows or column of x. Figure 6 illustrates an implementation for the fast two dimensional transformation of x.

The row-wise then column-wise transformation is not the only way the two-dimensional fast transform can be implemented. Rewriting the two-dimensional transform to look like a four-dimensional transform

$$\bar{x}\left(t\frac{N}{M}+u, v\frac{N}{M}+w\right) = \sum_{r=0}^{M-1} \sum_{s=0}^{M-1} e^{-2j\pi\frac{(tr+vs)}{M}} \left\{ e^{-2j\pi\frac{(ur+ws)}{N}} \left[\sum_{\ell=0}^{\frac{N}{M}-1} \sum_{k=0}^{\frac{N}{M}-1} e^{-2j\pi\frac{(u\ell+wk)}{N/M}} x(\ell M+r, kM+s) \right] \right\}$$

Now it appears that we pick every M^{th} point on a row and M^{th} point on a column and apply a two-dimensional transform on them. This amounts to taking

$$\left(\frac{N}{M}\right)^2$$

transforms. Then multiply through by twiddle factors and take M^2 more transforms on the corresponding points of the transform just taken. Figure 7 illustrates this implementation form for the two dimensional FFT.

Suppose now we permute the pixels of the 4 x 4 image, as shown in figure 8, and consider taking the two-dimensional FFT of the permuted image (implemented as in figure 7). What happens if instead of carrying through with the four layer FFT on the permuted image we only carry through for two layers? Figure 9 shows the resulting *incomplete fast transform on the permuted image*. It appears that the incomplete transform on the permuted image is identical to the transform obtained by partitioning the 4 x 4 image into 4 2 x 2 subimages and taking an FFT on each subimage. It should be clear from the form of equation (2) that this is the general situation for the Fourier transform. From the general form of the simple fast transform which is based on the direct vector product of basis vectors, from some basis sets, it should also be clear that an incomplete fast transform on the entire image corresponds to permuting the pixels (this allows the identity permutation), partitioning the image into subimages, and taking a complete transform on each subimage.

COMPOSITE MATRICES

The purpose of the transform coding compression techniques is the reduction of dimensionality while preserving data structure. When mean squared error is the criteria used in judging how well data structure is preserved, the optimal transform technique is the principal components, or by another name the Karhunen Loéve, (KL) transform. Assuming the image has zero mean, the basis vectors of this transformation are given by the eigenvectors of the autocovariance matrix for the set of *subimages into which the image is partitioned*. Thus in order to do a good job of data compression we must select a fast transform technique whose basis vectors are the eigenvectors of matrices similar to the form of the autocovariance matrix of the image. In this section of the paper we explore the form of the autocovariance matrix and the kinds of matrices which the fast transforms diagonalize.

We begin with some examples. Each of these examples is alike in the sense that the eigenvectors depend only on the form of the matrix and not on any of the values the elements of the matrix might take.

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}=\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} a+b & 0 \\ 0 & a-b \end{pmatrix}$$

$$\begin{pmatrix} a & b & c \\ b & a-b+c & b \\ c & b & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -2 \\ 1 & -1 & 1 \end{pmatrix}=\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -2 \\ 1 & -1 & 1 \end{pmatrix}\begin{pmatrix} a+b+c & 0 & 0 \\ 0 & a-c & 0 \\ 0 & 0 & a-2b+c \end{pmatrix}$$

$$\begin{pmatrix} a & b & c & d \\ d & a & b & c \\ c & d & a & b \\ b & c & d & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -i & j \end{pmatrix}=\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix}\begin{pmatrix} a+b+c+d & & & \\ & a+jb-c-jd & & \\ & & a-b+c-d & \\ & & & a-jb-c+jd \end{pmatrix}$$

Figure 10 illustrates some more examples.

Notice that in each of these examples, the eigenvalue corresponding to an eigenvector is easily obtained as the dot product of the eigenvector with the first row of the matrix e.g.

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/3 & -1/3 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}=\begin{pmatrix} a+b+c+d \\ a+b/3-c/3-d \\ a-b-c+d \\ a-3b+3c-d \end{pmatrix}$$

This occurs because the first component of each eigenvector is 1.

From this small set of 2nd, 3rd, and 4th order matrix-eigenvector forms we can construct larger order matrix-eigenvector forms in a way consistent with the fast transform technique. Consider for example a composite matrix of the form

$$A=\begin{pmatrix} H_1 & H_2 & H_3 \\ H_2 & H_1-H_2+H_3 & H_2 \\ H_3 & H_2 & H_1 \end{pmatrix}$$

where H_1, H_2, and H_3 all have the same eigenvectors (they commute) and are of the form

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix}$$

The eigenvectors of A are of the form $x \cdot y$, the direct product of

$$x=\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

an eigenvector of a matrix of the form

$$\begin{pmatrix} a & b & c \\ b & a-b+c & b \\ c & b & a \end{pmatrix},$$

with

$$y=\begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

an eigenvector of a matrix of the form

$$\begin{pmatrix} d & e \\ e & d \end{pmatrix}.$$

See Afriat, (1954), and Williamson, (1931). Why this must be so is easily illustrated in our example. Consider

$$A(x \cdot y)=\begin{pmatrix} H_1 & H_2 & H_3 \\ H_2 & H_1-H_2+H_3 & H_2 \\ H_3 & H_2 & H_1 \end{pmatrix}\begin{pmatrix} x_1y \\ x_2y \\ x_3y \end{pmatrix}$$

$$=\begin{pmatrix} x_1H_1y+x_2H_2y+x_3H_3y \\ x_1H_2y+x_2(H_1-H_2+H_3)y+x_3H_2y \\ x_1H_3y+x_2H_2y+x_3H_1y \end{pmatrix}$$

$$=\begin{pmatrix} x_1n_1y+x_2n_2y+x_3n_3y \\ x_1n_2y+x_2(n_1y-n_2y+n_3y)+x_3n_2y \\ x_1n_3y+x_2n_2y+x_3n_1y \end{pmatrix}$$

$$=\begin{pmatrix} x_1n_1+x_2n_2+x_3n_3 \\ x_1n_2+x_2(n_1-n_2+n_3)+x_3n_2 \\ x_1n_3+x_2n_2+x_3n_1 \end{pmatrix} \cdot y$$

$$= \cdot \left[\begin{pmatrix} n_1 & n_2 & n_3 \\ n_2 & n_1-n_2+n_3 & n_2 \\ n_3 & n_2 & n_1 \end{pmatrix} x \right] \cdot y$$

But since x is an eigenvector of any matrix of the form

$$\begin{pmatrix} a & b & c \\ b & a-b+c & b \\ c & b & a \end{pmatrix}, \qquad \begin{pmatrix} n_1 & n_2 & n_3 \\ n_2 & n_1-n_2+n_3 & n_2 \\ n_3 & n_2 & n_1 \end{pmatrix} x = \lambda x \ .$$

Hence,

$$A(x \cdot y) = [\lambda x] \cdot y = \lambda (x \cdot y).$$

Thus if y is an eigenvector of H_1, H_2, and H_3 so that $H_1 y = n_1 y$, $H_2 y = n_2 y$, and $H_3 y = n_3 y$ and x is an eigenvector of

$$B = \begin{pmatrix} n_1 & n_2 & n_3 \\ n_2 & n_1-n_2+n_3 & n_2 \\ n_3 & n_2 & n_1 \end{pmatrix},$$

the matrix B having the same form as A except that the corresponding eigenvalues appear in place of the H matrices, and x has corresponding eigenvalue λ, then $x \cdot y$ is an eigenvector of A, with eigenvalue λ.
In this example the eigenvectors of A are

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ -2 \\ -2 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ -2 \\ 2 \\ 1 \\ -1 \end{pmatrix}$$

and they correspond to the fast transform implemented as the direct product of the vectors in

$$\left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} \right\}$$

with those in

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\} .$$

The matrix A has the general form

$$\begin{pmatrix} a & b & c & d & e & f \\ b & a & d & c & f & e \\ c & d & a-c+e & b-d+f & c & d \\ d & c & b-d+f & a-c+e & d & c \\ e & f & c & d & a & b \\ f & e & d & c & b & a \end{pmatrix}$$

and has eigenvalues corresponding to the 6 listed eigenvectors given respectively by

(a+b+c+d+e+f), (a-b+c-d+e-f), (a+b-c-f)
(a-b-e+f), (a+b-2c-2d+e+f), and (a-b-2c+2d+e-f).

PRINCIPAL COMPONENTS TRANSFORM

The fast transform question is how close can we find a composite matrix to the general form of a autocovariance matrix. If we can find a composite matrix similar in form to the autocovariance matrix, then we would expect that the easily computed eigenvectors of the composite form would be similar to the eigenvectors of the autocovariance matrix and that the transformation of the image by these easily computed eigenvectors (which have a direct product form) can be implemented as a fast transform.

Figure 11 illustrates the general form for the 16 x 16 autocovariance matrix of a 4 x 4 image and table 1 lists which resolution cells in the 4 x 4 image the letters of the covariance matrix relate to. The partitioning of this matrix as 16 4 x 4 submatrices yields submatrices of two general forms. The first form is that of the diagonal 4 x 4 submatrices. The second form is that of the off-diagonal 4 x 4 submatrices, which appear as their own tranposes on one side of the diagonal. In general, these forms do not commute and the autocovariance matrix is not a composite matrix. Hence the eigenvectors cannot be expressed as a direct product of smaller dimensional vectors and there does not exist a simple fast transform implementation for them.

Under the assumption that the image is isotropic, the autocovariance matrix simplifies somewhat as illustrated in figure 12. This simplification might do some good since when the autocovariance matrix is partitioned into 4 x 4 submatrices all the matrices have the same form:

$$\begin{pmatrix} y & r & k & d \\ r & y & r & k \\ k & r & y & r \\ d & k & r & y \end{pmatrix}.$$

Unfortunately, unlike the example matrix-eigenvectors illustrated earlier, the eigenvectors for this matrix do depend on the values of the elements in addition to the form of the matrix. Thus although the autocovariance matrix for the isotropic image can be partitioned into matrices of the same form, these matrices do not, in general, have the same eigenvectors.

However, there still is enough structure in the matrix for us to capitalize on: the matrix is invariant under certain permutations of its rows and columns. The theory of permutation invariance is as follows. Suppose $Ax = \lambda x$ so that x is an eigenvector of A and λ is its corresponding eigenvalue. Suppose P is an permutation matrix and that A is invariant under P; that is, A = P'AP. Now consider the eigenvectors of P'AP. Since x was an eigenvector of A and A = P'AP we must also have $P'APx = \lambda x$. But for a permutation operator $P^{-1} = P'$ so that A (Px) = (Px). Hence, Px must also be an eigenvector of A having corresponding eigenvalue λ.

There are two permutations under which the autocovariance matrix of the 4 x 4 isotropic image is invariant. They are

$$P_1 = \begin{pmatrix} 0000000000000001 \\ 0000000000000010 \\ 0000000000000100 \\ 0000000000010000 \\ 0000000000100000 \\ 0000000001000000 \\ 0000000010000000 \\ 0000000100000000 \\ 0000001000000000 \\ 0000010000000000 \\ 0000100000000000 \\ 0001000000000000 \\ 0010000000000000 \\ 0100000000000000 \\ 1000000000000000 \end{pmatrix} \qquad P_2 = \begin{pmatrix} 0001000000000000 \\ 0010000000000000 \\ 0100000000000000 \\ 1000000000000000 \\ 0000000100000000 \\ 0000001000000000 \\ 0000010000000000 \\ 0000100000000000 \\ 0001000000000000 \\ 0010000000000000 \\ 0100000000000000 \\ 1000000000000000 \\ 0000000000000001 \\ 0000000000000010 \\ 0000000000000100 \\ 0000000000001000 \end{pmatrix}$$

Both P_1 and P_2 are symmetric and are their own inverses. The autocovariance matrix is invariant under P_1 because it is symmetric about both diagonals. It is invariant under P_2 because each of the 4 x 4 submatrices of the autocovariance matrix is symmetric about both diagonals.

P_1 and P_2 will tell us something about the form of the eigenvectors. We will consider P_1 first. Suppose x is an eigenvector of the autocovariance matrix

$$x' = (x_1, x_6, \ldots, x_{16}).$$

Then $P_1 x$ is also an eigenvector of the autocovariance matrix. Writing x and $P_1 x$ with first component 1, we can set them equal since they both have the same corresponding eigenvalue. Hence

$$(1, x_2/x_1, x_3/x_1 \ldots, x_{16}/x_1)' =$$

$$(1, x_{15}/x_{16}, x_{14}/x_{16} \ldots, x_1/x_{16})'.$$

Notice that

$$\frac{x_{16}}{x_1} = \frac{x_1}{x_{16}}.$$

This implies that

$$x_{16} = \pm x_1.$$

Taking

$$x_{16} = x_1,$$

we see that

$$x_n = x_{17-n}$$

Taking

$$x_{16} = -x_1,$$

we see that

$$x_n = -x_{17-n}.$$

Thus the eigenvectors have two forms: even and odd. And the kinds of matrices they diagonalize must be bisymmetric (Shanmugam & Haralick, 1973).

In our analysis of the invariance of the eigenvector to P_2 we need only consider the vector y of the first eight components. Compare y and y with its first four and last four components reversed. Notice that

$$\frac{y_4}{y_1} = \frac{y_1}{y_4}$$

so that

$$y_4 = \pm y_1.$$

When

$$y_4 = y_1$$

we get the form

$$(1, y_2/y_1, y_2/y_1, 1, y_5/y_1, y_6/y_1, y_6/y_1, y_5/y_1)$$

When

$$y_4 = -y_1$$

we get the form

$$(1, y_2/y_1, -y_2/y_1, -1, y_5/y_1, y_6/y_1, -y_6/y_1, -y_5/y_1)$$

This implies that there are four sets of eigenvectors, each set having four vectors of the form

$$\begin{pmatrix} a \\ b \\ b \\ a \\ e \\ f \\ f \\ e \\ e \\ f \\ f \\ e \\ a \\ b \\ b \\ a \end{pmatrix} \quad \begin{pmatrix} a \\ b \\ -b \\ -a \\ e \\ f \\ -f \\ -e \\ -e \\ -f \\ f \\ e \\ -a \\ -b \\ b \\ a \end{pmatrix} \quad \begin{pmatrix} a \\ b \\ b \\ a \\ e \\ f \\ f \\ e \\ -e \\ -f \\ -f \\ -e \\ -a \\ -b \\ -b \\ -a \end{pmatrix} \quad \begin{pmatrix} a \\ b \\ -b \\ -a \\ e \\ f \\ -f \\ -e \\ e \\ f \\ -f \\ -e \\ a \\ b \\ -b \\ -a \end{pmatrix}$$

Figure 13 illustrates the fastest way to implement the dot product of a vector with each of the four eigenvectors in each set. Compared with the general 16 x 1 fast transform which requires 128 add and multiply operations, the implementation shown for the eigenvector of the 16 x 16 autocovariance matrix requires 192 operations.

Our next question must be: how generalizable are these results? In the general isotropic case for the N row by M column subimage, the autocovariance matrix can be partitioned into N^2 M x M submatrices each of the same Toeplitz form. Although the Toeplitz form is the same, the eigenvectors depend on the values in the Toeplitz matrix form as well as on the form itself. Hence, we cannot say that the N^2 submatrices have the same eigenvectors.

If we are willing to make one more idealization - that the submatrices differ by a multiplicative constant - then that would imply that they all have the same eigenvectors. To say that the submatrices are proportional is to say that whatever covariance relationships a row has to itself, the covariance relationship of one row with another row is the same modulo a multiplicative constant. It is an assumption not as strong as the 1st order Markov dependence assumption often made. Now the existence of a fast transform follows from our previous discussion of composite matrices. The number of operations involved in the fast implementation would be $N \cdot M^2$ for the first layer and $M \cdot N^2$ for the second layer. This is a total of $(M+N)NM$ operations compared to $(NM)^2$ for a brute force implementation.

CONCLUSIONS

We have discussed the general fast transform from the perspective of the vector direct product and have shown the relationship between data compression transform coding and the incomplete fast transform. We have illustrated that the autocovariance matrix for an isotropic 4 x 4 image yields eigenvectors which have a semi-fast transform. We have discussed what assumption beyond isotropy is needed for the fast KL transform to exist in the general case. In a future paper we will discuss the suitability of these additional assumptions.

REFERENCES

S. N. Afriat, "Composite Matrices," Quarterly Journal of Mathematics, Oxford, Vol. 2, No. 5, 1954, pp. 81-98.

I. J. Good, "The Relationship Between Two Fast Fourier Transforms," IEEE C-20, March, 1971, pp. 310-317.

R. M. Haralick and K. Shanmugam, "Comparative Study of a Discrete Linear Basis for Image Data Compression," IEEE SMC-4, No. 1, Jan., 1974, pp. 16-27.

K. Shanmugam and R. M. Haralick, "A Computationally Simple Procedure for Imagery Data Compression by the Karhunen-Loéve Method," IEEE SMC-3, No. 2, March, 1973, pp. 202-204.

J. Williamson, "The Latent Roots of a Matrix of Special Type," Bulletin Amer. Math. Society, Vol. 37, 1931, pp. 585-90.

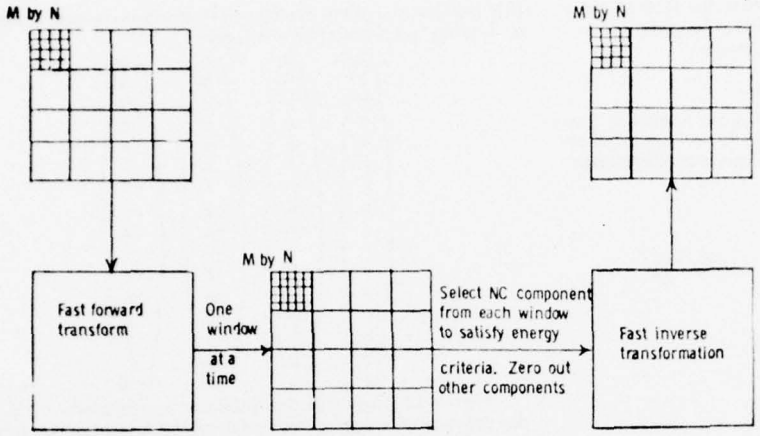


Figure 1 The transform coding technique

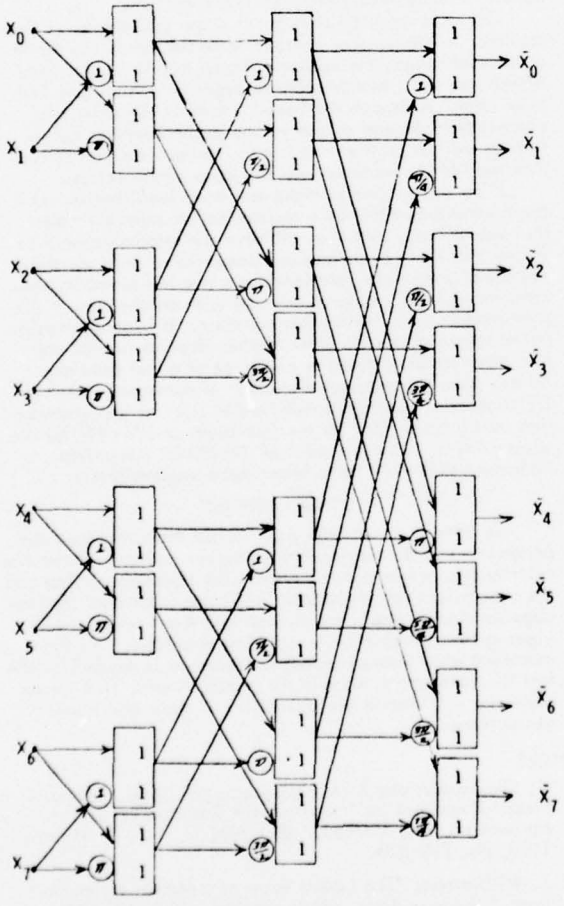


Figure 3 Fast Fourier Transform

Figure 2 Illustration of the fast implementation of the transform defined by the direct product of vectors in S_1 and those in S_2

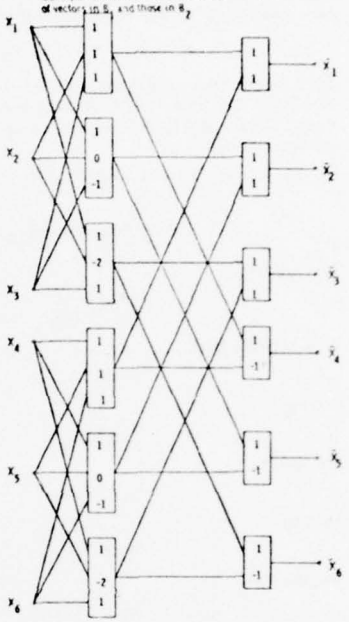


Figure 4 is placed under Figure 13.

X(0,0)	X(0,1)	X(0,2)	X(0,3)
X(1,0)	X(1,1)	X(1,2)	X(1,3)
X(2,0)	X(2,1)	X(2,2)	X(2,3)
X(3,0)	X(3,1)	X(3,2)	X(3,3)

Figure 5 A 4 by 4 image

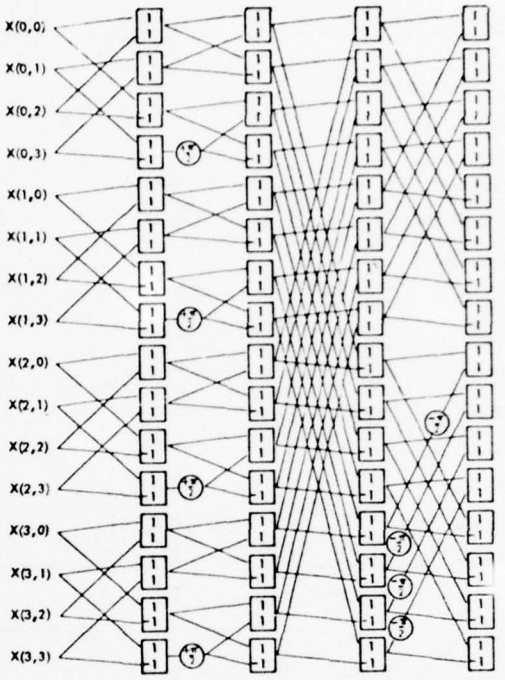


Figure 6. Shows a fast FFT for the 4 X 4 image.

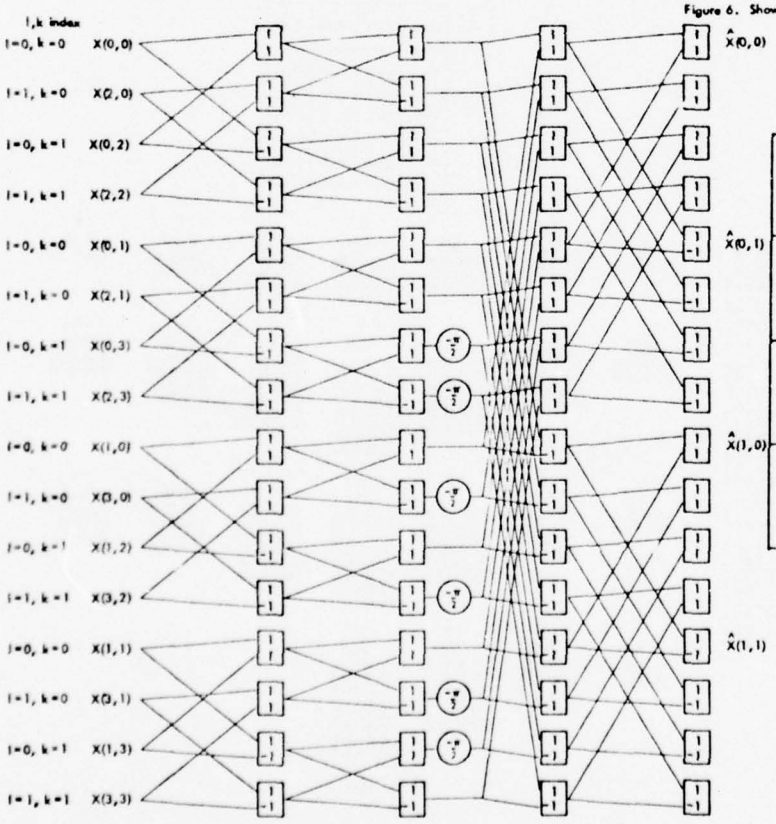


Figure 7. Shows another way to implement an FFT on the 4 X 4 image.

X(0,0)	X(0,2)	X(0,1)	X(0,3)
X(2,0)	X(2,2)	X(2,1)	X(2,3)
X(1,0)	X(1,2)	X(1,1)	X(1,3)
X(3,0)	X(3,2)	X(3,1)	X(3,3)

Figure 8
A permutated image 4 by 4

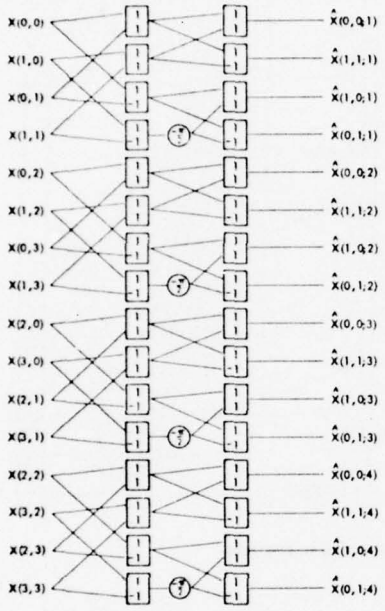


Figure 9. Shows an incomplete two-dimensional FFT implemented on the permutated image of figure 8.

$$\begin{pmatrix} a & b & c & d \\ b & a & d & c \\ c & d & a & b \\ d & c & b & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+d-c-d \\ a-b+c-d \\ a-b-c+d \end{pmatrix}$$

$$\begin{pmatrix} a & b & c & d \\ b & a-\frac{4b}{3}+\frac{4c}{3} & \frac{4b}{3}-\frac{4c}{3}+d & c \\ c & \frac{4b}{3}-\frac{4c}{3}+d & a-\frac{4b}{3}+\frac{4c}{3} & b \\ d & c & b & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/3 & -1 & -3 \\ 1 & -1/3 & -1 & 3 \\ 1 & -1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/3 & -1 & -3 \\ 1 & -1/3 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+\frac{b}{3}-\frac{c}{3}-d \\ a-b-c+d \\ a-3b+3c-d \end{pmatrix}$$

$$\begin{pmatrix} a & d & b & c \\ b & a & c & d \\ d & c & a & b \\ c & b & d & a \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+jb-c-jd \\ a-b+c-d \\ a-jd+c+jd \end{pmatrix}$$

$$\begin{pmatrix} a & c & d & b \\ c & a & b & d \\ b & d & a & c \\ d & b & c & a \end{pmatrix}\begin{pmatrix} 1 & 1 & +1 & 1 \\ 1 & -1 & +1 & -1 \\ 1 & j & -1 & +j \\ 1 & -j & -1 & +j \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & +j \\ 1 & -j & -1 & +j \end{pmatrix}\begin{pmatrix} a+b+c+d \\ a+jb-c-jd \\ a-b+c-d \\ a-jb-c+jd \end{pmatrix}$$

$$\begin{pmatrix} q & r & s & t & u \\ r & q-t+\frac{s+u}{2} & \frac{s+t}{2} & u-t+\frac{s+r}{2} & t \\ s & \frac{s+t}{2} & q+u-s & \frac{s+t}{2} & s \\ t & u-t+\frac{s+r}{2} & \frac{s+t}{2} & q-t+\frac{s+u}{2} & r \\ u & t & s & r & q \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 & +1 & 1 \\ 1 & 1/2 & 0 & -2 & -3/2 \\ 1 & 0 & -2 & 0 & 1 \\ 1 & -1/2 & 0 & -2 & -3/2 \\ 1 & -1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1/2 & 0 & -2 & -3/2 \\ 1 & 0 & -2 & 0 & 1 \\ 1 & -1/2 & 0 & -2 & -3/2 \\ 1 & -1 & 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} q+r+s+t+u \\ q+1/2r-1/2t-u \\ q-2s+u \\ q-2r-2t-u \\ q-\frac{3r}{2}+s-\frac{3t}{2}+u \end{pmatrix}$$

Figure 10. Examples of diagonalized matrices.

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

Resolution Cell	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)	(4,1)	(4,2)	(4,3)	(4,4)
(1,1)	y	x	w	v	r	q	p	o	k	j	i	h	d	c	b	a
(1,2)	x	y	x	w	t	r	q	p	l	k	j	i	e	d	c	b
(1,3)	w	x	y	x	t	s	r	q	m	l	k	j	f	e	d	c
(1,4)	v	w	x	y	u	t	s	r	n	m	l	k	g	f	e	d
(2,1)	r	s	t	u	y	x	w	v	r	q	p	o	k	j	i	h
(2,2)	q	r	s	t	x	y	x	w	s	r	q	p	l	k	j	i
(2,3)	p	q	r	s	w	x	y	x	t	s	r	q	m	l	k	j
(2,4)	o	p	q	r	v	w	x	y	u	t	s	r	n	m	l	k
(3,1)	k	l	m	n	r	s	t	u	y	x	w	v	r	q	p	o
(3,2)	j	k	l	m	q	r	s	t	x	y	x	w	s	r	q	p
(3,3)	i	j	k	l	p	q	r	s	w	x	y	x	t	s	r	q
(3,4)	h	i	j	k	o	p	q	r	v	w	x	y	u	t	s	r
(4,1)	d	c	f	g	k	l	m	n	r	s	t	u	y	x	w	v
(4,2)	c	d	e	f	j	k	l	m	q	r	s	t	x	y	x	w
(4,3)	b	c	d	e	i	j	k	l	p	q	r	s	w	x	y	x
(4,4)	a	b	c	d	h	i	j	k	o	p	q	r	v	w	x	y

Figure 11. General form autocovariance matrix (16 X 16)

a (3,3): (1,1)-(4,4)

b (3,2): (1,1)-(4,3), (1,2)-(4,4)

c (3,1): (1,1)-(4,2), (1,2)-(4,3), (1,3)-(4,4)

d (3,0): (1,1)-(4,1), (1,2)-(4,2), (1,3)-(4,3), (1,4)-(4,4)

e (3,-1): (1,2)-(4,1), (1,3)-(4,2), (1,4)-(4,3)

f (3,-2): (1,3)-(4,1), (1,4)-(4,2)

g (3,-3): (1,4)-(4,1)

h (2,3): (1,1)-(3,4), (2,1)-(4,4)

i (2,2): (1,1)-(3,3), (1,2)-(3,4), (2,1)-(4,3), (2,2)-(4,4)

j (2,1): (1,1)-(3,2), (1,2)-(3,3), (1,3)-(3,4), (2,1)-(4,2), (2,2)-(4,3), (2,3)-(4,4)

k (2,0): (1,1)-(3,1), (1,2)-(3,2), (1,3)-(3,3), (1,4)-(3,4), (2,1)-(4,1), (2,2)-(4,2), (2,3)-(4,3), (2,4)-(4,4)

l (2,-1): (1,2)-(3,1), (1,3)-(3,2), (1,4)-(3,3), (2,2)-(4,1), (2,3)-(4,2), (2,4)-(4,3)

m (2,-2): (1,3)-(3,1), (1,4)-(3,2), (2,3)-(4,1), (2,4)-(4,2)

n (2,-3): (1,4)-(3,1), (2,4)-(4,1)

o (1,3): (1,1)-(2,4), (2,1)-(3,4)

p (1,2): (1,1)-(2,3), (1,2)-(2,4), (2,1)-(3,3), (2,2)-(3,4)

q (1,1): (1,1)-(2,2), (1,2)-(2,3), (1,3)-(2,4), (2,1)-(3,2), (2,2)-(3,3), (2,3)-(3,4)

r (1,0): (1,1)-(2,1), (1,2)-(2,2), (1,3)-(2,3), (1,4)-(2,4), (2,1)-(3,1), (2,2)-(3,2), (2,3)-(3,3), (2,4)-(3,4)

s (1,-1): (1,2)-(2,1), (1,3)-(2,2), (1,4)-(2,3), (2,2)-(3,1), (2,3)-(3,2), (2,4)-(3,3)

t (1,-2): (1,3)-(2,1), (1,4)-(2,2), (2,3)-(3,1), (2,4)-(3,2)

u (1,-3): (1,4)-(2,1), (2,4)-(3,1), (3,4)-(4,1)

v (0,3): (1,1)-(1,4), (2,1)-(2,4)

w (0,2): (1,1)-(1,3), (1,2)-(1,4), (2,1)-(2,3), (2,2)-(2,4)

x (0,1): (1,1)-(1,2), (1,2)-(1,3), (1,3)-(1,4), (2,1)-(2,2), (2,2)-(2,3), (2,3)-(2,4)

y (0,0): (1,1)-(1,1), (1,2)-(1,2), (1,3)-(1,3), (1,4)-(1,4), (2,1)-(2,1), (2,2)-(2,2), (2,3)-(2,3), (2,4)-(2,4)

TABLE 1

Resolution Cell	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)	(4,1)	(4,2)	(4,3)	(4,4)
(1,1)	y	r	k	d	r	q	j	c	k	j	i	b	d	c	b	a
(1,2)	r	y	r	k	q	r	q	j	j	k	j	i	c	d	c	b
(1,3)	k	r	y	r	j	q	r	q	i	j	k	j	b	c	d	c
(1,4)	d	k	r	y	c	j	q	r	b	i	j	k	a	b	c	d
(2,1)	r	q	j	c	y	r	k	d	r	q	j	c	k	j	i	b
(2,2)	q	r	q	j	r	y	r	k	q	r	q	j	j	k	j	i
(2,3)	j	q	r	q	k	r	y	r	j	q	r	q	i	j	k	j
(2,4)	c	j	q	r	d	k	r	y	c	j	q	r	b	i	j	k
(3,1)	k	j	i	b	r	q	j	c	y	r	k	d	r	q	j	c
(3,2)	j	k	j	i	q	r	q	j	r	y	r	k	q	r	q	j
(3,3)	i	j	k	j	j	q	r	q	k	r	y	r	j	q	r	q
(3,4)	b	i	j	k	c	j	q	r	d	k	r	y	c	j	q	r
(4,1)	d	c	b	a	k	j	i	b	r	q	j	c	y	r	k	d
(4,2)	c	d	c	b	j	k	j	i	q	r	q	j	r	y	r	k
(4,3)	b	c	d	c	i	j	k	j	j	q	r	q	k	r	y	r
(4,4)	a	b	c	d	b	i	j	k	c	j	q	r	d	k	r	y

Figure 12. Isotropic autocovariance form

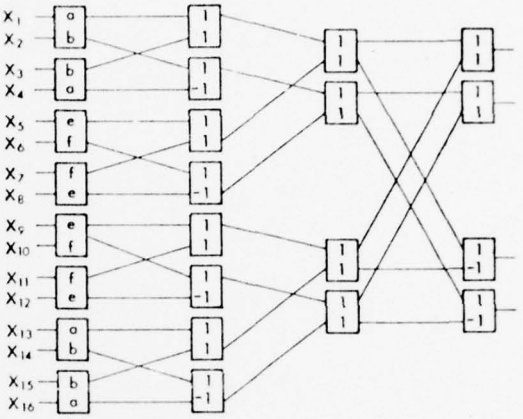


Figure 13 Fast implementation for Eigenvectors

ACKNOWLEDGMENT

This work was supported in part by the Air Force Avionics Laboratory, Wright-Patterson Air Force Base Ohio, U. S. Air Force contract number F33615-74-C-1093.

R. M. Haralick is with the Remote Sensing Lab and the Department of Electrical Engineering, University of Kansas, Lawrence, Kansas, 66045.

N. Griswold is with the Air Force Avionics Laboratory presently working at the Remote Sensing Lab , University of Kansas, 66045.

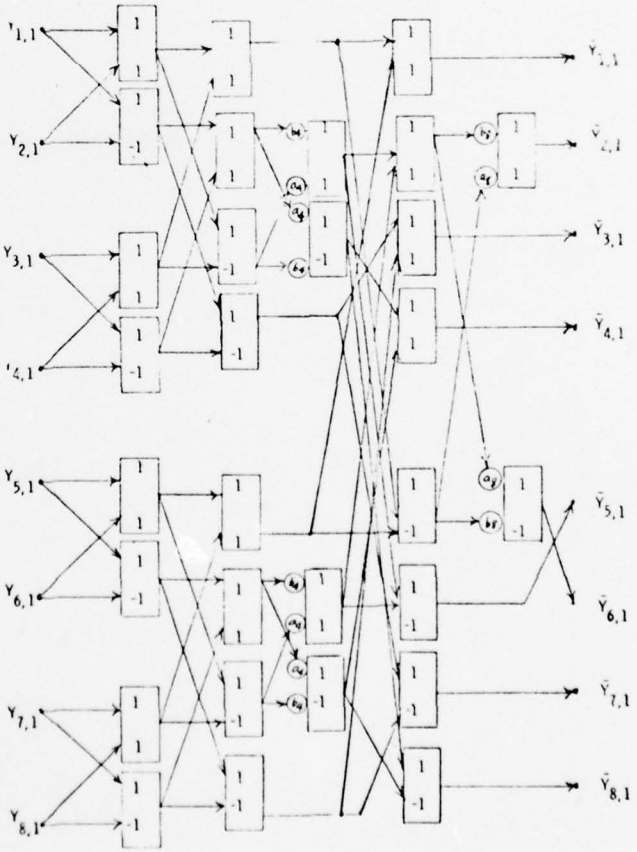


Figure 4 The Slant transform

APPENDIX II

Derivation of Optimum Variable Word Length Code

Given that a total number of projections are to be considered (NPR) the truncation error, by using only p projections, where p projections (or coefficients) are selected by first ordering the variances of the total projections, where $p \leq NPR$ is:

$$T = \sum_{I=1}^{NPR} VAR(I) - \sum_{I=1}^{p} VAR(I) \qquad (1)$$

where $VAR(I)$ = Variance of the I^{th} component

$$= E\ \{(x_I - m_I)^2\}$$

with x_I is the I^{th} data component

m_I is the mean of the I^{th} component

The total quantization error for the image is given by:

$$Q = \sum_{I=1}^{p} VAR(I)\ C^{-2\ Nbits(I)} \qquad (2)$$

For p Joel Max quantizers

$C = 1.78$ for normal distribution of any one component

$Nbits$ = bit assignment for the I^{th} component.

Combining equation (1) and (2) the total system error is:

$$E = \underbrace{\sum_{I=1}^{NPR} VAR(I) - \sum_{I=1}^{p} VAR(I)}_{\text{Truncation Error}} + \underbrace{\sum_{I=1}^{p} VAR(I)\ \left(C^{-2\ Nbits(I)}\right)}_{\text{min. variance quantization error}}$$

$$\text{or } E = L - \sum_{I=1}^{p} VAR(I)\ \left(1 - C^{-2\ Nbits(I)}\right) \qquad (3)$$

The total error is therefore a function of p, the number of projections retained and Nbits, the number of bits assigned to the p^{th} quantizer. The truncation error decreases with increasing p, while the quantization error increases with increasing p. It is necessary to find the value of p and resulting bit assignments for the set of $\{Nbits(I)\}_{I=1}^{p}$ which minimizes the total error E.

Given that some total number of bits, say Mbits is to be assigned to p Joel Max quantizers

$$\text{i.e.} \quad \sum_{I-1}^{p} Nbits(I) = Mbits$$

determine the optimum number of projections p (ordered according to decreasing variances) and corresponding bit assignments of $\{Nbits(I)\}_{I=1}^{p}$ that minimizes the total system error

$$E = T + Q$$

Restated in a better form:

$$\min_{(p,\ Nbits(I))} \left\{ L - \sum_{I=1}^{p} VAR(I) \left(1 - C^{-2\ Nbits(I)}\right) \right\}$$

where $L = \sum_{I=1}^{NPR} VAR(I)$ and by definition is a positive constant

Under the constraint $\sum_{I=1}^{p} Nbits(I) = Mbits$

Since L is a constraint:

$$L \geq \sum_{I=1}^{p} VAR(I) \left(1 - C^{-2\ Nbit(I)}\right)$$

with the equality p = NPR and Mbits → ∞. An equivalent problem is to maximize the equation on the right or:

$$\max_{(p,\ \mathrm{Nbits}(I))} \left\{ \sum_{I=1}^{p} \mathrm{VAR}(I) \left(1 - C^{-2\ \mathrm{Nbits}(I)}\right) \right\}$$

$$\text{Subject to } \sum_{I=1}^{p} \mathrm{Nbits}(I) = \mathrm{Mbits}$$

Considering Nbits and p projections as continous variables and using Lagrange multipliers λ we have

$$\frac{\partial}{\partial\, \mathrm{Nbits}\,(J)} \left[\sum_{I=1}^{p} \mathrm{VAR}(I) \left(1 - C^{-2\ \mathrm{Nbits}(I)}\right) + \lambda \sum_{I=1}^{p} \mathrm{Nbits}(I) \right] = 0 \Bigg\}_{J=1}^{p} \quad (4)$$

$$\frac{\partial}{\partial p} \left\{ \sum_{I=1}^{p} \mathrm{VAR}(I) \left(1 - C^{-2\ \mathrm{Nbits}(I)}\right) + \lambda \sum_{I=1}^{p} \mathrm{Nbits}(I) \right\} = 0 \quad (5)$$

$$\sum_{I-1}^{p} \mathrm{Nbits}(I) = \mathrm{Mbits} \quad (6)$$

Equation (4) represents p equations while (5) and (6) represent one equation each. There is then p + 2 equations and p + 2 unknowns

$$\left(p,\ \lambda,\ \left\{\mathrm{Nbits}(I)\right\}_{I=1}^{p}\right)$$

Taking the partial derivations of (4) and noting

$$e^{\ln\ C^{-2\mathrm{Nbits}(I)}} = C^{-2\mathrm{Nbits}(I)}$$

we have $\mathrm{VAR}(J)\ (2)\ \ln\ C^{-2\mathrm{Nbits}(J)} + \lambda = 0 \quad (7)$

likewise if (5) is approximated by

$$\frac{\partial}{\partial p} \left\{ \int_1^P VAR(I) \left(1 - C^{-2Nbits(I)}\right) dI + \lambda \int_1^P Nbits(I)\, dI \right\} = 0$$

by fundamental theorem of Calculus:

$$\left\{ VAR(p) \left(1 - C^{-2Nbits(P)}\right) + \lambda\, Nbits(p) \right\} = 0$$

$$\text{or } VAR(p) \left(1 - C^{-2Nibts(p)}\right) + \lambda\, Nbits(p) = 0 \qquad (8)$$

Solving equation (7) for Nbits

$$2\, VAR(J)\, \ln\, C^{-2Nbit(J)} + \lambda = 0$$

$$\ln C^{-2Nibts(J)} = \frac{-\lambda}{2\, VAR(J)}$$

Taking $\log_C$ we have:

$$\log_C \left[\ln C^{-2Nbits(J)}\right] = \log_C \left[\frac{-\lambda}{2\, VAR(J)}\right]$$

$$2\, Nbits(J) \log_C \left[\ln C\right] = -\log_C \left[\frac{-\lambda}{2\, VAR(J)}\right]$$

$$2\, Nbits(J) = \frac{-\log_C \left[\frac{-\lambda}{2\, VAR(J)}\right]}{\log_C \left[\ln C\right]} = -\log_C \left[\frac{-\lambda}{2\, VAR(J) \ln c}\right]$$

$$Nbits(J) = -1/2 \log_C \left[\frac{-\lambda}{2\, VAR(J) \ln C}\right] = -1/2 \log_C \left[\frac{-\lambda}{2\, \ln C\ VAR(J)}\right]$$

$$= 1/2 \log_C \left[\frac{-\lambda}{2\, \ln C\, VAR(J)}\right]^{-1}$$

$$\text{Nbits}(J) = 1/2 \log_C \left[\frac{-2 \; n \; C \; \text{VAR}(J)}{\lambda}\right] \tag{9}$$

for $A = \ln C$

$$\text{Nbits}(J) = 1/2 \log_C \left[\frac{-2 \; A \; \text{VAR}(J)}{\lambda}\right]$$

Substituting this result into (6)

$$\text{Mbits} = \sum_{I=1}^{P} \text{Nbits}(J)$$

$$\text{Mbits} = \sum_{I=1}^{P} 1/2 \log_C \left[\frac{-2 \; A \; \text{VAR}(J)}{\lambda}\right]$$

$$= 1/2 \log_C \left\{\left[\frac{-2A}{\lambda}\right]^P \prod_{j=1}^{P} \text{VAR}(J)\right\} \tag{10}$$

Now we must solve (10) for λ

$$\left\{\left[\frac{-2A}{\lambda}\right]^P \prod_{I=1}^{P} \text{VAR}(I)\right\} = C^{2\text{Mbits}}$$

$$\left[\frac{-2A}{\lambda}\right]^P = \frac{C^{2\,\text{Mbits}}}{\prod_{I-1}^{P} \text{VAR}(I)}$$

$$\left[\frac{-2A}{\lambda}\right] = \left[\frac{C^{2\text{Mbits}}}{\prod_{I=1}^{P} \text{VAR}(I)}\right]^{1/p}$$

$$\lambda = -2A \left[\frac{C^{2\text{Mbits}}}{\prod_{I-1}^{P} \text{VAR}(I)}\right]^{P}$$

Substituting into (9)

$$Nbits(J) = 1/2 \log_C \left[\frac{-2A \; VAR(J)}{-2A \left[\frac{C^{2Mbits}}{\prod_{I=1}^{p} VAR(I)} \right]^{p}} \right]$$

$$Nbits(J) = 1/2 \log_C \left[VAR(J) \left[\frac{C^{2Mbits}}{\prod_{I=1}^{p} VAR(I)} \right]^{1/p} \right] \qquad (10a)$$

This result shows that:

$$Nbits(J) = 1/2 \log_C (VAR(J)) + \frac{Mbits}{p} - \frac{1}{2p} \log_C \left[\prod_{I=1}^{p} VAR(I) \right]$$

$$Nbits(J) = 1/2 \log_C VAR(J) + \frac{Mbits}{p} - \frac{1}{2p} \sum_{I=1}^{p} \log_C VAR(I)$$

$$Nbits(J) = 1/2 \left[\frac{\log_C VAR(J) \ln C}{\ln C} \right] + \frac{Mbits}{p} - \frac{1}{2p} \sum_{I=1}^{p} \frac{\log_C VAR(I) \ln C}{\ln C}$$

but $\log_C VAR(I) \ln C = \ln VAR(I)$

$$Nbits(J) = \frac{1}{2A} \ln VAR(J) + \frac{Mbits}{p} - \frac{1}{2Ap} \sum_{I=1}^{p} \ln VAR(I) \qquad (11)$$

Evaluating at J = p and substituting into (8) repeating Eq. (8)

$$VAR(p) - VAR(p) \, C^{-2Nbits(p)} + \lambda \, Nbits(p) = 0$$

substituting for last turn on left

$$- \left[\frac{C^{2Mbits}}{\prod_{I=1}^{p} VAR(I)} \right]^{p} \left[\ln VAR(p) + \frac{2AMbits}{p} - \frac{1}{p} \sum_{I=1}^{p} \ln VAR(I) \right]$$

$$- \left[\frac{\prod_{I=1}^{p} VAR(I)}{C^{2Mbits}} \right]^{1/p} \left[\ln VAR(p) + \frac{2A\,Mbits}{p} - \frac{1}{p} \sum_{I=1}^{p} \ln VAR(I) \right]$$

Substituting for second term on left of Eq. (8)

$$C^{-2Nbits(p)} = \left[VAR(p) \left[\frac{C^{2Mbits}}{\prod_{I=1}^{p} VAR(I)} \right]^{1/p} \right]^{-1}$$

$$= \frac{VAR(p)}{VAR(p) \left[\frac{C^{2Mbits}}{\prod_{I=1}^{p} VAR(I)} \right]^{-1/p}}$$

Resulting in $\left[\frac{\prod_{I=1}^{p} VAR(I)}{C^{2Mbits}} \right]^{1/p}$

Eq. (8) then becomes:

$$VAR(p) - \underbrace{\left[\frac{\prod_{I=1}^{p} VAR(I)}{C^{2Mbits}} \right]^{1/p}}_{\text{term 1}} \underbrace{\left[1 + \ln VAR(p) + \frac{2A\,Mbits}{p} - \frac{1}{p} \sum_{I=1}^{p} \ln VAR(I) \right]}_{\text{term 2}} \quad (12)$$

The value of p satisfying Eq. (12) and bit assignment by (11) represent the approximate solution to the minimization described earlier.

Now given the N x N transformation T, the transform vector y is created by the transform encoder. The number of bits/vector element (the rate R) is specified thus:

$$Mbits = N \cdot R$$

Eq. (12) is then used to determine the value p which most closely satisfies the equality

$$NOPT = \min_{1 \leq p \leq NPR} \left\{ \left| VAR(p) - \left[\frac{\prod_{I=1}^{p} VAR(I)}{C^{2Mbits}} \right]^{1/p} \left[1 + \ln (VAR(p)) + \frac{2A\,Mbits}{p} - \frac{1}{p} \sum_{I=1}^{p} \ln VAR(I) \right] \right| \right\} \quad (13)$$

The values of the Nbits(I) generated are not necessarily integers. The integers must therefore be chosen by some rule. This was accomplished by rounding off each Nbits(I) to the nearest integer. If the resulting rate R exceeds or is less then that specified then one bit is added or subtracted, one at a time for minimum error. It is this rounding off procedure which will be replaced by the Huffman Coder.

APPENDIX III

APPENDIX III

Theorem 2

Let $x_1, x_2, \ldots, x_K$ be given vectors in R^N. Let $P: R^N \to V$ be an orthogonal projection operator onto the subspace V_M, spanned by the M eigenvectors of $\sum_{k=1}^{K} x_k x_k'$ with largest eigenvalues. Let $\varepsilon_1^2 = \sum_{k=1}^{K} (x_k - Px_k)'(x_k - Px_k)$. Let $z = \frac{1}{K} \sum_{k=1}^{K} x_k$. Let $P^*: R^N \to V^*$ be an orthogonal projection operator onto the subspace V^* spanned by the M eigenvectors of $\sum_{k=1}^{K} (x_k - z)(x_k - z)'$ with largest eigenvalues. Let $\varepsilon_2^2 = \sum_{k=1}^{K} ((x_k - z) - P^*(x_k - z))'((x_k - z) - P^*(x_k - z))$. Then $\varepsilon_1^2 \geq \varepsilon_2^2$.

Proof: Simplifying ε_1^2 and ε_2^2 we obtain

$$\varepsilon_1^2 = \text{Trace } ((I-P) \sum_{k=1}^{K} x_k x_k')$$

$$\varepsilon_2^2 = \text{Trace } ((I-P^*) \sum_{k=1}^{K} (x_k - z)(x_k - z)')$$

But, $\sum_{k=1}^{K} (x_k - z)(x_k - z)' = \sum_{k=1}^{K} x_k x_k' - \sum_{k=1}^{K} x_k z' - z \sum_{k=1}^{K} x_k + Kzz'$

$$= \sum_{k=1}^{K} x_k x_k' - Kzz'.$$

Hence, $\varepsilon_1^2 = \text{Trace } \left((I-P) \left(\sum_{k=1}^{K} (x_k - z)(x_k - z)' + Kzz' \right) \right)$

$$\varepsilon_2^2 = \text{Trace} \left((I-P^*) \left(\sum_{k=1}^{K} (x_k-z)(x_k-z)' \right) \right)$$

From principal components we must have

$$\text{Trace} \left((I-P^*) \sum_{k=1}^{K} (x_k-z)(x_k-z)' \right) \leq \text{Trace} \left((I-P) \sum_{k=1}^{K} (x_k-z)(x_k-z)' \right)$$

Therefore, $\varepsilon_2^2 \leq \varepsilon_1^2 - \text{Trace}((I-P)Kzz')$, or

$$\varepsilon_2^2 \leq \varepsilon_1^2 - Kz'(I-P)z.$$

Since $(I-P)$ is positive semi-definite, $z'(I-P)z \geq 0$. Consequently, $\varepsilon_2^2 \leq \varepsilon_1^2 - Kz'(I-P)z \leq \varepsilon_1^2$.

Theorem 2 has shown that if we translate the data by its mean, the squared error for an M-dimensional projection is less than without any translation. We now show that no other translation is better than the mean.

Theorem 3

Let $x_1, x_2, \ldots, x_k$ be given vectors in R^N. Let $P_a: R^N \to R^N$ be the orthogonal projection operator of rank M which minimizes $\varepsilon^2(a) = \sum_{k=1}^{K} (x_k-a)'(I-P_a)(x_k-a)$. Then, $\varepsilon^2(z) \leq \varepsilon^2(a)$ where

$$z = \frac{1}{K} \sum_{k=1}^{K} x_k.$$

Proof: consider $\varepsilon^2(a)$,

$$\varepsilon^2(a) = \sum_{k=1}^{K} (x_k-a+z-z)'(I-P_a)(x_k-a+z-z)$$

$$= \sum_{k=1}^{K} (x_k-z)'(I-P_a)(x_k-z)+2(x_k-z)'(I-P_a)(z-a)+(z-a)'(I-P_a)(z-a)$$

But $\sum_{k=1}^{K} (x_k-z)=0$; hence,

$$\varepsilon^2(a) = \sum_{k=1}^{K}(x_k-z)'(I-P_a)(x_k-z)+ K(z-a)'(I-P_a)(z-a)$$

Now $\varepsilon^2(z) = \sum_{k=1}^{K} (x_k-z)'(I-P_z)(x_k-z)$. By principal components,

$\sum_{k=1}^{K}(x_k-z)'(I-P_z)(x_k-z) \leq \sum_{k=1}^{K} (x_k-z)'(I-P_a)(x_k-z)$. Therefore,

$\varepsilon^2(z) \leq \varepsilon^2(a)-K(z-a)'(I-P_a)(z-a)$. Since $(I-P_a)$ is positive semi-definite, $(z-a)'(I-P_a)(z-a) \geq 0$. Consequently,

$\varepsilon^2(z) \leq \varepsilon^2(a)$.

BIBLIOGRAPHY

Anderson, T. W., An Introduction to Multivariate Statistical Analysis, John Wiley and Sons, Inc., New York, 1958.

Andrews, Harry C., Computer Techniques in Image Processing, Academic Press, New York, 1970.

Brigham, E. Oran, The Fast Fourier Transform, Prentice-Hall, New Jersey, 1974.

Gold and Rader, Digital Processing of Signals, McGraw-Hill, New York, 1969.

Goodman, J. W., Introduction to Fourier Optics, McGraw-Hill, New York, 1968.

Hogg, Robert V., and Craig, Allen T., Introduction to Mathematical Statistics, The Macmillan Co., New York, 1970.

Huang, Thomas S., and Tretiak, Ohl J., Picture Bandwidth Compression, Gordon and Breach, New York, 1972.

Morrison, Donald F., Multivariate Statistical Methods, McGraw-Hill, Inc., New York, 1967.

Morrison, Norman, Introduction to Sequential Smoothing and Prediction, McGraw-Hill, Inc., New York, 1969.

Reza, Fazlollah M., An Introduction to Information Theory, McGraw-Hill Book Co., New York, 1961.

ARTICLES

Afrait, S. N., "Composite Matrices," Quarterly Journal of Mathematics, Oxford, Vol. 2, No. 5, 1954, pp. 81-88.

Agarwal, Ramesh, C. and Burrus, Sidney C., "Number Theoretic Transforms to Implement Fast Digital Convolution," Proceedings IEEE, Vol. 63, No. 4, April 1975.

Ahmed, N., Natarajan, T., and Rao, K. R. "Discrete Cosine Transform", IEEE Transactions on Computers, January 1974.

Anderson, G., and Huang, T. S., "Piecewise Fourier Transformation for Picture Bandwidth Compression", IEEE Transactions on Communications Techniques, Vol. COM-19, No. 2, April 1971.

Andrews, Harry C., "Multidimensional Rotations in Feature Selection", IEEE Transactions, Vol. C-20, No. 9, Sept. 1971.

Andrews, Harry C., "N Topics in Search of an Editorial: Heuristics, Superresolution and Bibliography", Proceedings of IEEE, Vol. 60, No. 7, July 1972, pp. 891-894.

Andrews, Tescher and Kruger, "Image Processing by Digital Computer", IEEE Spectrum, Vol. 9, No. 7, July 1972, pp. 20-32.

Arguello, Roger, "Encoding, Transmission, and Decoding of Sampled Images," Prepared for Perkin-Elmer Corp., 1971.

Arguello, Sellner and Stuller, "The Effect of Channel Errors in the Differential Pulse-Code-Modulation Transmission of Sampled Imagery", IEEE Transactions on Communication Tech., Dec. 1971.

Arguello, Sellner and Stuller, "Transfer Function Compensation of Sampled Imagery", IEEE Transactions on Computers, Vol. C-21, No. 7, July 1972, pp. 812-818.

Bennett, W. R., "Spectra of Quantized Signals", Bell System Tech. Journal

Billingsley, F. C., "Computer-Generated Color Image Display of Lunar Spectral Reflectance Ratios", Photographic Science and Engineering, Vol. 16, No. 1, Jan.-Feb. 1972, pp. 51-57.

Good, I. J., "The Relationship Between Two Fast Fourier Transforms," IEEE C-20, March 1971., pp. 310-317.

Habibi, Ali, "Hybrid Coding of Pictoral Data", IEEE Transactions on Communication, Vol. COM-22, No. 5, May 1974.

Haralick, R. M., and Shanmugam, K., "Comparative Study of a Discrete Linear Basis for Image Data Compression", IEEE SMC, Vol SMC-4, No. 1, Jan. 1974, pp. 16-27.

Haralick, R. M. and Griswold, N., "Image Data Compression: The Incomplete Fast Transform", IEEE SMC-CHO 908-4, Oct. 1974, pp. 299-306.

Haralick, R. M. and Shanmugam, K., "A Computationally Simple Procedure for Imagery Data Compression by the Karhunen Loeve Method", IEEE SMC-3, No. 2, March 1973, pp. 202-204.

Haralick, R. M., Griswold, N., Kattiyakulwanich, N., "A Fast Two Dimensional Karhunen Loeve Transform", SPIE, August 1975.

Haralick, R. M., Shanmugam, K., and Goel, D., "Telemetry Data Compression Using Image Compression Techniques", Cadre Corporation, Lawrence, Kansas, August 1974.

Max, Joel, "Quantizing for Minimum Distortion", IRE Transactions on Information Theory, March 1960.

NASA Tech. Brief: "Data Compression by Decreasing Slop Threshold Test", NASA, Pasadena Office, September 1973.

O'Handley, Douglas and Green, "Recent Developments in Digital Image Processing at the Image Processing Laboratory at the Jet Propulsion Laboratory", Proc. of the IEEE, Vol. 60, No. 7, July 1972.

Oppenheim Alan et al., "Nonlinear Filtering of Multiplied and Convolved Signals", Proc. of the IEEE, Vol. 56, No. 8, August 1968.

Papoulis, A., "Limits on Bandlimited Signals", Proceedings of the IEEE, Vol. 55, No. 10, October 1967.

Pearl, Andrews and Pratt, "Performance Measures for Transform Data Coding", IEEE Transactions on Communications, Vol. COM-20, No. 3, June 1972, pp. 411-415.

Pratt, William K., "Vector Formulation of Two-Dimensional Signal Processing Operations", Computer Graphics and Image Processing, 1975.

Pratt, William K., Kane, J., and Andrews, H. C., "Hadamard Transform Image Coding", Proceedings of the IEEE, Vol. 37, No. 1, January 1969.

Rabiner, Lawrence, "Techniques for Designing Finite-Duration Impulse-Response Digital Filters", IEEE Transactions on Communication Tech., Vol. COM-19, No. 2, April 1971.

Sawchuk, Alexander, "Space-Variant Image Motion Degradation and Restoration", Proc. of the IEEE.

Schaming, W. B., "Digital Image Transform Encoding", RCA, PE G22, 1974.

Sondhi, Man Mohan, "Image Resotration: The Removal of Spatially Invariant Degradiations", Proc. of the IEEE, Vol. 60, No. 7, July 1972.

Stockham, T. G., "Image Processing in the Context of a Visual Model", Proceedings of the IEEE, Vol. 60, No. 7, July 1972, pp. 828-842.

Tasto and Wintz, "Image Coding by Adaptive Block Quantization", IEEE Transactions on Communication Tech., Vol. COM-19, No. 6, December 1971.

Williamson, L., "The Latent Roots of a Matrix of Special Type", Bulletin American Mathematical Society, Vol. 37, 1931, pp. 585-590.

Wintz, Paul A., "Transform Picture Coding", Proceedings of the IEEE, Vol. 60, No. 7, July 1972, pp. 809-820.

UNPUBLISHED PAPERS

Arguello et al., "Statistics of Aerial Imagery", Produced for Perkin-Elmer Corp., Norwalk, Conn., Presented at Symposium on Image Processing, Purdue University, April 30, 1973.

Billingsley, F. C., "Image Processing for Electron Microscopy: A Digital System", Prepared at Jet Propulsion Lab, CIT, Under NASA Contract No. NAS 7-100.

Billingsley, F. C., "Noise Considerations in System Design", Presented at the Symposium on Image Processing at Purdue University, April 31, 1973.

Billingsley and Goetz, "Computer Techniques Used for Some Enhancements of ERTS Images", Prepared under Contract No. NAS 7-100, sponsored by NASA and Presented at the ERTS-1 Investigator's Symposium March 5-9, 1973.

Fu and Swain, "On Syntactic Pattern Recognition", Prepared under USDA Contract No. 12-14-100-9549(50) and NASA Contract No. NGR 15-005-112; Presented at the Purdue University Symposium on Image Processing, April 30, 1973.

Haralick, R. M., "Principal Components Analysis", CRES Technical Memorandum 177-5, April 1970. Remote Sensing Laboratory, University of Kansas Center for Research, Lawrence, Kansas 66044.

Lackey, Robert, "So What's A Walsh Function", Produced by the EE Department, Ohio State University, Columbus.

Landgrebe, David, "Systems Approach to the Use of Remote Sensing", Presented at the Symposium on Image Processing at Purdue University, April 30, 1973.

Nahi and Assefi, "Bayesian Recursive Image Estimation", Prepared for Air Force Eastern Test Range under Contract No. F0806-72-C-0008 under NASA Multidisciplinary Research Grant NGL-05-018-044.

Nathan, Robert, "Image Processing for Electron Microscopy: Enhancement Procedures", Prepared at the Jet Propulsion Lab, CIT, Under NASA Contract No. NAS 7-100.

Stockham, Thomas, "Intra-Frame Encoding for Monochrome Images by Means of a Psychophysical Model Based on Nonlinear Filtering of Multiplied Signals", Presented at the Symposium on Image Processing at Purdue University, April 30, 1973.

Tamburino and Michaels, "RPV Systems Engineering for High Probability of Acquisition with Anti-Jam Bandwidth Constraint", Prepared by AFAL/AAI Control Data Retrieval Systems Working Group, WPAFB, March 30, 1973.

REFERENCES

SECTION I

1. Wintz, Paul A., "Transform Picture Coding", Proc. of the IEEE, Vol. 60, No. 7, July 1972, pp. 809-820.

2. Good, I. J., "The Interaction Algorithm and Practical Fourier Analysis", Journal of Royal Statistical Society (London) B-20, 361, 1958.

3. Cooley, J. W. and Tukey, J. W., "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. of Comput., Vol. 19, pp. 297-301, April 1965.

4. Andrews, H. C., Pratt, W. K., "Fourier Transform Coding of Images," in 1969 Int. Conf. System Sciences, Honolulu, Hawaii, pp. 677-679.

5. Anderson, Grant and Huang, "Piecewise Fourier Transformation for Picture Bandwidth Compression", IEEE Transactions on Communications Techniques, Vol. com-19, No. 2, April 1971.

6. Andrews, Harry, Computer Techniques in Image Processing, Academic Press, New York, 1970.

7. Lackey, Robert, "So What's A Walsh Function", Produced by the EE Dept., Ohio State University, Columbus, Ohio.

8. Pratt, William K., Kane, and Andrews, "Hadamard Transform Image Coding", Proc. of the IEEE, Vol. 37, No. 1, Jan., 1969.

9. Haralick, Shanmugam, Goel and Young, "A Comparative Study of Transform Data Compression Techniques for Digital Imagery", National Electronics Conference, Chicago, October, 1972,

10. Enomoto, H., and Shibutu, K., "Orthogonal Transform Coding System for Television Signals", Journal of the Institute of TV Engineers of Japan, Vol. 24, No. 2, February 1970, pp. 99-108.

11. Pratt, W. et al., "Slant Transforms for Image Coding", Proceedings of the IEEE, March, 1972, On Applications of Walsh Functions, Washington, D. C.

12. Ahmed, N., Natarajan, T., and Rao, K. R., "Discrete Cosine Transform", IEEE Transactions on Computers, January, 1974.

13. Shanmugam, K. Sam, "Comments on Discrete Cosine Transforms", IEEE Transactions on Computers, July, 1974.

14. Haralick, R. M., "A Storage Efficient Way to Implement the Discrete Cosine Transform", Tech. note (to be published), Remote Sensing Laboratory, University of Kansas, June, 1975.

15. Max, Joel, "Quantizing for Minimum Distortion", IRE Transactions on Information Theory, March, 1960.

16. Habibi, Ali, "Comparison of the nth-Order DPCM Encoder with Linear Transformations and Block Quantization Techniques", IEEE Transactions on Communication Techniques, Vol. com-19, No. 6, December 1971.

17. Tasto and Wintz, "Image Coding by Adaptive Block Quantization", IEEE Transactions on Communication Techniques, Vol. com-19, No. 6, December 1971.

18. Connor, D. J. et al., "Interframe Coding for Picture Transmission", Proceedings of the IEEE, July, 1972, pp. 779-790.

19. Huang, Thomas S., and Tretiak, Ohl, J., Picture Bandwidth Compression, Gordon and Breach, New York, 1972.

20. Pratt, W. K. et al., "Semiannual Technical Report", Image Processing Institute, University of Southern Calif., Los Angeles, Calif., ARPA Order No. 1706, Contract No. FO606-72-C-0008.

21. Haralick, Young, Goel and Shanmugam, "A Comparative Study of Data Compression Techniques for Digital Image Transmission", Cadre Corporation, Lawrence, Kansas, Technical Report, February 1972.

22. Habibi, Ali, "Hybrid Coding of Pictorial Data", IEEE Transactions on Communications, Vol. com-22, No. 5, May 1974.

23. Andrews, H. C. and Patterson, C. L., "Outer Product Expansions and Their Uses in Digital Image Processing", American Mathematical Monthly, Vol. 1, No. 82, January 1975, pp. 1-13.

24. Haralick, R. M., Shanmugam, K., and Goel, D., "Telemetry Data Compression Using Image Compression Techniques", Cadre Corporation, Lawrence, Kansas, August 1974.

SECTION II

1. S. N. Afrait, "Composite Matrices," Quarterly Journal of Mathematics, Oxford, Vol. 2, No. 5, 1954, pp. 81-88.

2. J. Williamson, "The Latent Roots of a Matrix of Special Type," Bulletin Amer. Math. Society, Vol. 37, 1931, pp. 585-590.

3. K. Shanmugam and R. M. Haralick, "A Computationally Simple Procedure for Imagery Data Compression by the Karhunen-Loeve Method," IEEE SMC-3, No. 2, March 1973, pp. 202-204.

4. R. M. Haralick and N. Griswold, "Image Data Compression: The Incomplete Fast Transform," IEEE SMC-CHO 908-4, Oct. 2-4, 1974, pp. 299-306.

5. R. M. Haralick and K. Shanmugam, "Comparative Study of A Discrete Linear Basis for Image Data Compression," IEEE SMC-4, No. 1, Jan. 1974, pp. 16-27.

SECTION III

1. Andrews, Harry, Computer Techniques in Image Processing, Academic Press, New York, 1970.

2. Max, J., "Quantization for Minimum Distortion," IRE Transactions on Information Theory, March, 1970.

3. Ready, P. and P. Wintz, "LARS Information Note 05072", Purdue University, March, 1972.

4. Hogg, R. V. and A. T. Craig, "Introduction to Mathematical Statistics," Macmillan Company, 1970, 375 pp.